
EffiPy Documentation

Release 0.6.1

GISCE-TI, S.L.

Sep 27, 2017

Contents

1	Argot	3
2	EffiPy API	5
2.1	Resource	5
2.2	EffiPeople service	6
2.3	Customers	7
2.4	Contracts	9
2.5	Invoices	10
2.6	Tariffs	11
2.7	UsagePoints	13
2.8	Meters	14

EffiPy is a pythonic library to access EffiPeople REST API based on [libsaas](#)

CHAPTER 1

Argot

- Create is `service.object().create(dict)`.
- Read all is `service.object().get()`.
- Read one is `service.object(identifier).get()`.
- Update all is `service.object().update(list_of_dicts)`.
- Update one is `service.object(identifier).update(dict)`.
- Delete is `service.object(identifier).delete()`.

Note: All the `get()` methods have optional parameters `limit` and `offset`.

Resource

```
class effipy.resource.EffiPeopleResource (parent, object_id=None)
```

```
    Bases: libsaas.services.base.RESTResource
```

```
    add_filter (filter_function)
```

```
    apply_filters (request)
```

```
    create (*args, **kwargs)
```

```
        Create a new resource.
```

Variables **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

```
    delete (*args, **kwargs)
```

```
        Delete this resource.
```

```
    filters = ()
```

```
    get (*args, **kwargs)
```

```
    get ()
```

```
    `get()` method for all the EffiPeople resources.
```

Parameters

- **start** (*int*) – Start of the results
- **limit** (*int*) – Limit of the results

```
    get_url ()
```

```
        Get the URL for the endpoint.
```

```
        In our case, we replace '/' for '-' for the object identifier.
```

```
list_methods ()
list_resources ()
parent = None
path = None
require (condition)
require_collection ()
require_item ()
update (*args, **kwargs)
wrap_object (obj)
```

EffiPeople service

```
class effipy.service.EffiPeople (token, version='v1', test=False)
```

Bases: `libsaas.services.base.Resource`

EffiPeople Service API.

```
add_filter (filter_function)
```

```
apply_filters (request)
```

```
customer (customer_id)
```

Specific Customer in `effipy.customers.Customers`.

```
>>> from effipy import EffiPeople
>>> ef = EffiPeople(token='XXXXX')
```

Customers

```
>>> ef.customer(1234).get_url()
'https://api.effipeople.com/v1/customers/1234'
```

Contracts

```
>>> ef.customer(1234).contracts().get_url()
'https://api.effipeople.com/v1/customers/1234/contracts'
>>> ef.customer(1234).contract(1).get_url()
'https://api.effipeople.com/v1/customers/1234/contracts/1'
>>> ef.customer(1234).contract(1).versions().get_url()
'https://api.effipeople.com/v1/customers/1234/contracts/1/versions'
>>> ef.customer(1234).contract(1).version(1).get_url()
'https://api.effipeople.com/v1/customers/1234/contracts/1/versions/1'
```

Invoices

```
>>> ef.customer(1234).contract(1).invoices().get_url()
'https://api.effipeople.com/v1/customers/1234/contracts/1/invoices'
>>> ef.customer(1234).contract(1).invoice(1).get_url()
'https://api.effipeople.com/v1/customers/1234/contracts/1/invoices/1'
```

```
customers ()
```

Endpoint to Customers in `effipy.customers.Customers`.

```
>>> from effipy import EffiPeople
>>> ef = EffiPeople(token='XXXXX')
>>> ef.customers().get_url()
'https://api.effipeople.com/v1/customers'
```

efficiency_reports()

filters = ()

get_url()

list_methods()

list_resources()

measure(measure_id)

measures()

parent = None

require(condition)

tariff(tariff_id)

Endpoint to Tariff *effipy.tariffs.Tariffs*.

```
>>> from effipy import EffiPeople
>>> ef = EffiPeople(token='XXXXX')
>>> ef.tariff('T1').get_url()
'https://api.effipeople.com/v1/tariffs/T1'
>>> ef.tariff('T1').versions().get_url()
'https://api.effipeople.com/v1/tariffs/T1/versions'
>>> ef.tariff('T1').version('V1').get_url()
'https://api.effipeople.com/v1/tariffs/T1/versions/V1'
```

tariffs()

usagepoint(point_id)

Endpoint to specific UsagePoint *effipy.usagepoints.Usagepoints*.

```
>>> from effipy import EffiPeople
>>> ef = EffiPeople(token='XXXXX')
>>> ef.usagepoint('ES00001111').get_url()
'https://api.effipeople.com/v1/usagepoints/ES00001111'
```

usagepoints()

Endpoint to UsagePoints.

```
>>> from effipy import EffiPeople
>>> ef = EffiPeople(token='XXXXX')
>>> ef.usagepoints().get_url()
'https://api.effipeople.com/v1/usagepoints'
```

use_json(request)

Customers

```
class effipy.customers.Customers(parent, object_id=None)
    Bases: effipy.resource.EffiPeopleResource
```

Customers management

Use `object_id` to get the endpoint to a specific customer.

create (*dict*)

Create a new customer with a dict that will be converted to JSON.

Parameters **dict** (*dict*) – A dictionary with customer data like the following:

```
{
  "id": "Identificador unico cliente",
  "fiscalId": "CIF, NIF, Pasaporte",
  "firstName": "Nombre",
  "firstSurname": "Primer apellido",
  "secondSurname": "Segundo apellido",
  "email": "correo@cliente.com",
  "address": {
    "street": "Calle y número",
    "postalCode": "CodigoPostal",
    "city": "Nombre ciudad",
    "cityCode": "Código INE ciudad",
    "province": "Nombre provincia",
    "provinceCode": "Código INE provincia",
    "country": "España",
    "countryCode": "Codigo",
    "parcelNumber": "Referencia catastral"
  }
}
```

add_filter (*filter_function*)

apply_filters (*request*)

contract (*object_id*)

contracts ()

delete (**args, **kwargs*)

Delete this resource.

filters = ()

get (**args, **kwargs*)

get ()

``get()` method for all the EffiPeople resources.`

Parameters

- **start** (*int*) – Start of the results
- **limit** (*int*) – Limit of the results

get_url ()

Get the URL for the endpoint.

In our case, we replace `'/'` for `'-'` for the object identifier.

list_methods ()

list_resources ()

parent = None

```

path = 'customers'
require (condition)
require_collection ()
require_item ()
update (*args, **kwargs)
wrap_object (obj)

```

Contracts

class `effipy.contracts.Contracts` (*parent*, *object_id=None*)

Bases: `effipy.resource.EffiPeopleResource`

Permite gestionar los contratos de los clientes de la compañía.

create (*dict*)

Create a new contract with a dict that will be converted to JSON.

```
>>> ef.customer(1234).contracts().create()
```

Parameters *dict* (*dict*) – A dictionary with contract data like the following:

```

{
    "id": "IdentificadorContrato",
    "customerId": "IdentificadorCliente",
    "version": "2",
    "start": 1332806400,
    "end": 1362009600,
    "tariffId": "2.0A",
    "power": 3300,
    "activityCode": "CNAE",
    "usagePointId": "ES0987543210987654ZF"
}

```

add_filter (*filter_function*)

apply_filters (*request*)

delete (**args*, ***kwargs*)

Delete this resource.

filters = ()

get (**args*, ***kwargs*)

get ()

`get()` method for all the EfficPeople resources.

Parameters

- **start** (*int*) – Start of the results
- **limit** (*int*) – Limit of the results

```
get_url ()
    Get the URL for the endpoint.

    In our case, we replace '/' for '-' for the object identifier.

invoice (object_id)

invoices ()

list_methods ()

list_resources ()

parent = None

path = 'contracts'

require (condition)

require_collection ()

require_item ()

update (*args, **kwargs)

version (object_id)

versions ()

wrap_object (obj)
```

Invoices

```
class effipy.invoices.Invoices (parent, object_id=None)
```

Bases: *effipy.resource.EffiPeopleResource*

```
create (dict)
```

Create a new invoice with a dict that will be converted to JSON.

Parameters **dict** (*dict*) – A dictionary with invoice data like the following:

```
{
    "number": "Número de factura",
    "customerId": "Identificador único del cliente",
    "contractId": "Identificador único del contrato",
    "usagePointId": "ES0987543210987654ZF",
    "activityCode": "Código CNAE",
    "tariffId": "2.0A",
    "power": 5500,
    "activeEnergyConsumption": 400,
    "reactiveEnergyConsumption": 225,
    "isConsumptionEstimated": false,
    "activeEnergyCost": 200.0,
    "reactiveEnergyCost": 100.0,
    "powerCost": 300.0,
    "servicesCost": 75.0,
    "rentalsCost": 100.0,
    "otherCost": 50.0,
    "subtotal": 825.0,
    "otherTaxes": 6.75,
    "currencyCode": "Euro",
    "invoiceDate": 1363301230,
```

```

    "periodStart": 1363301020,
    "periodEnd": 13633012000,
    "vat": 173.25,
    "totalTaxes": 180.0,
    "total": 1005.0,
    "tariffType": "2.0DHA"
}

```

add_filter (*filter_function*)

apply_filters (*request*)

delete (**args, **kwargs*)

Delete this resource.

filters = ()

get (**args, **kwargs*)

get ()

``get()`` method for all the EffiPeople resources.

Parameters

- **start** (*int*) – Start of the results
- **limit** (*int*) – Limit of the results

get_url ()

Get the URL for the endpoint.

In our case, we replace ‘/’ for ‘-’ for the object identifier.

list_methods ()

list_resources ()

measure (*measure_id*)

measures ()

parent = None

path = ‘invoices’

require (*condition*)

require_collection ()

require_item ()

update (**args, **kwargs*)

wrap_object (*obj*)

Tariffs

class `effipy.tariffs.Tariffs` (*parent, object_id=None*)

Bases: `effipy.resource.EffiPeopleResource`

Permite gestionar las tarifas de la compañía.

create (*dict*)

Create a new tariffs with a dict that will be converted to JSON.

Parameters **dict** (*dict*) – A dictionary with tariffs data like the following:

```
{
  "id": "2.0DHA",
  "name": "Luz 10 (2.0A)",
  "tariffType": "2.0DHA",
  "version": "2",
  "start": 1363301230,
  "end": 1363304568,
  "periods": {
    "p1": {
      "activeEnergyPrice": 0.167658,
      "powerPrice": 1.824432
    },
    "p2": {
      "activeEnergyPrice": 0.05719,
      "powerPrice": 1.824432
    }
  }
}
```

add_filter (*filter_function*)

apply_filters (*request*)

delete (**args, **kwargs*)

Delete this resource.

filters = ()

get (**args, **kwargs*)

get ()

`get()` method for all the EffiPeople resources.

Parameters

- **start** (*int*) – Start of the results
- **limit** (*int*) – Limit of the results

get_url ()

Get the URL for the endpoint.

In our case, we replace ‘/’ for ‘-’ for the object identifier.

list_methods ()

list_resources ()

parent = None

path = ‘tariffs’

require (*condition*)

require_collection ()

require_item ()

```

update (*args, **kwargs)
version (object_id)
versions ()
wrap_object (obj)

```

UsagePoints

class `effipy.usagepoints.UsagePoints` (*parent*, *object_id=None*)

Bases: `effipy.resource.EffiPeopleResource`

create (*dict*)

Create a new Usagepoint with a dict that will be converted to JSON.

Parameters *dict* (*dict*) – A dictionary with Usagepoint data like the following:

```

{
  "id": "ES0987543210987654ZF",
  "address": {
    "street": "Calle y número",
    "postalCode": "CodigoPostal",
    "city": "Nombre ciudad",
    "cityCode": "Código INE ciudad",
    "province": "Nombre provincia",
    "provinceCode": "Código INE provincia",
    "country": "España",
    "countryCode": "ES. Codigo según ISO 3166",
    "parcelNumber": "Referencia catastral"
  }
}

```

add_filter (*filter_function*)

apply_filters (*request*)

delete (*args, **kwargs)

Delete this resource.

filters = ()

get (*args, **kwargs)

get ()

``get()`` method for all the EfficPeople resources.

Parameters

- **start** (*int*) – Start of the results
- **limit** (*int*) – Limit of the results

get_url ()

Get the URL for the endpoint.

In our case, we replace `'/'` for `'-'` for the object identifier.

list_methods ()

```
list_resources ()
meter (object_id)
meters ()
parent = None
path = 'usagepoints'
require (condition)
require_collection ()
require_item ()
update (*args, **kwargs)
wrap_object (obj)
```

Meters

```
class effipy.meters.Meters (parent, object_id=None)
    Bases: effipy.resource.EffiPeopleResource
```

Permite gestionar los contadores que realizan las medidas

```
create (dict)
```

Create a new invoice with a dict that will be converted to JSON.

Parameters **dict** (*dict*) – A dictionary with invoice data like the following:

```
{
    "id": "sample string 1",
    "usagePointId": "sample string 2"
}
```

```
add_filter (filter_function)
```

```
apply_filters (request)
```

```
delete (*args, **kwargs)
```

Delete this resource.

```
filters = ()
```

```
get (*args, **kwargs)
```

```
get ()
```

``get()`` method for all the **EffiPeople** resources.

Parameters

- **start** (*int*) – Start of the results
- **limit** (*int*) – Limit of the results

```
get_url ()
```

Get the URL for the endpoint.

In our case, we replace `'/'` for `'-'` for the object identifier.

```
list_methods ()
list_resources ()
parent = None
path = 'meters'
reading (reading_datetime, period=None)
readings ()
require (condition)
require_collection ()
require_item ()
update (*args, **kwargs)
wrap_object (obj)
```


A

`add_filter()` (effipy.contracts.Contracts method), 9
`add_filter()` (effipy.customers.Customers method), 8
`add_filter()` (effipy.invoices.Invoices method), 11
`add_filter()` (effipy.meters.Meters method), 14
`add_filter()` (effipy.resource.EffiPeopleResource method), 5
`add_filter()` (effipy.service.EffiPeople method), 6
`add_filter()` (effipy.tariffs.Tariffs method), 12
`add_filter()` (effipy.usagepoints.UsagePoints method), 13
`apply_filters()` (effipy.contracts.Contracts method), 9
`apply_filters()` (effipy.customers.Customers method), 8
`apply_filters()` (effipy.invoices.Invoices method), 11
`apply_filters()` (effipy.meters.Meters method), 14
`apply_filters()` (effipy.resource.EffiPeopleResource method), 5
`apply_filters()` (effipy.service.EffiPeople method), 6
`apply_filters()` (effipy.tariffs.Tariffs method), 12
`apply_filters()` (effipy.usagepoints.UsagePoints method), 13

C

`contract()` (effipy.customers.Customers method), 8
`Contracts` (class in effipy.contracts), 9
`contracts()` (effipy.customers.Customers method), 8
`Contracts.create()` (built-in function), 9
`Contracts.get()` (built-in function), 9
`create()` (effipy.resource.EffiPeopleResource method), 5
`customer()` (effipy.service.EffiPeople method), 6
`Customers` (class in effipy.customers), 7
`customers()` (effipy.service.EffiPeople method), 6
`Customers.create()` (built-in function), 8
`Customers.get()` (built-in function), 8

D

`delete()` (effipy.contracts.Contracts method), 9
`delete()` (effipy.customers.Customers method), 8
`delete()` (effipy.invoices.Invoices method), 11
`delete()` (effipy.meters.Meters method), 14

`delete()` (effipy.resource.EffiPeopleResource method), 5
`delete()` (effipy.tariffs.Tariffs method), 12
`delete()` (effipy.usagepoints.UsagePoints method), 13

E

`efficiency_reports()` (effipy.service.EffiPeople method), 7
`EffiPeople` (class in effipy.service), 6
`EffiPeopleResource` (class in effipy.resource), 5
`EffiPeopleResource.get()` (built-in function), 5

F

`filters` (effipy.contracts.Contracts attribute), 9
`filters` (effipy.customers.Customers attribute), 8
`filters` (effipy.invoices.Invoices attribute), 11
`filters` (effipy.meters.Meters attribute), 14
`filters` (effipy.resource.EffiPeopleResource attribute), 5
`filters` (effipy.service.EffiPeople attribute), 7
`filters` (effipy.tariffs.Tariffs attribute), 12
`filters` (effipy.usagepoints.UsagePoints attribute), 13

G

`get()` (effipy.contracts.Contracts method), 9
`get()` (effipy.customers.Customers method), 8
`get()` (effipy.invoices.Invoices method), 11
`get()` (effipy.meters.Meters method), 14
`get()` (effipy.resource.EffiPeopleResource method), 5
`get()` (effipy.tariffs.Tariffs method), 12
`get()` (effipy.usagepoints.UsagePoints method), 13
`get_url()` (effipy.contracts.Contracts method), 9
`get_url()` (effipy.customers.Customers method), 8
`get_url()` (effipy.invoices.Invoices method), 11
`get_url()` (effipy.meters.Meters method), 14
`get_url()` (effipy.resource.EffiPeopleResource method), 5
`get_url()` (effipy.service.EffiPeople method), 7
`get_url()` (effipy.tariffs.Tariffs method), 12
`get_url()` (effipy.usagepoints.UsagePoints method), 13

I

`invoice()` (effipy.contracts.Contracts method), 10

Invoices (class in `effipy.invoices`), 10
`invoices()` (`effipy.contracts.Contracts` method), 10
`Invoices.create()` (built-in function), 10
`Invoices.get()` (built-in function), 11

L

`list_methods()` (`effipy.contracts.Contracts` method), 10
`list_methods()` (`effipy.customers.Customers` method), 8
`list_methods()` (`effipy.invoices.Invoices` method), 11
`list_methods()` (`effipy.meters.Meters` method), 14
`list_methods()` (`effipy.resource.EffiPeopleResource` method), 5
`list_methods()` (`effipy.service.EffiPeople` method), 7
`list_methods()` (`effipy.tariffs.Tariffs` method), 12
`list_methods()` (`effipy.usagepoints.UsagePoints` method), 13
`list_resources()` (`effipy.contracts.Contracts` method), 10
`list_resources()` (`effipy.customers.Customers` method), 8
`list_resources()` (`effipy.invoices.Invoices` method), 11
`list_resources()` (`effipy.meters.Meters` method), 15
`list_resources()` (`effipy.resource.EffiPeopleResource` method), 6
`list_resources()` (`effipy.service.EffiPeople` method), 7
`list_resources()` (`effipy.tariffs.Tariffs` method), 12
`list_resources()` (`effipy.usagepoints.UsagePoints` method), 13

M

`measure()` (`effipy.invoices.Invoices` method), 11
`measure()` (`effipy.service.EffiPeople` method), 7
`measures()` (`effipy.invoices.Invoices` method), 11
`measures()` (`effipy.service.EffiPeople` method), 7
`meter()` (`effipy.usagepoints.UsagePoints` method), 14
Meters (class in `effipy.meters`), 14
`meters()` (`effipy.usagepoints.UsagePoints` method), 14
`Meters.create()` (built-in function), 14
`Meters.get()` (built-in function), 14

P

`parent` (`effipy.contracts.Contracts` attribute), 10
`parent` (`effipy.customers.Customers` attribute), 8
`parent` (`effipy.invoices.Invoices` attribute), 11
`parent` (`effipy.meters.Meters` attribute), 15
`parent` (`effipy.resource.EffiPeopleResource` attribute), 6
`parent` (`effipy.service.EffiPeople` attribute), 7
`parent` (`effipy.tariffs.Tariffs` attribute), 12
`parent` (`effipy.usagepoints.UsagePoints` attribute), 14
`path` (`effipy.contracts.Contracts` attribute), 10
`path` (`effipy.customers.Customers` attribute), 8
`path` (`effipy.invoices.Invoices` attribute), 11
`path` (`effipy.meters.Meters` attribute), 15
`path` (`effipy.resource.EffiPeopleResource` attribute), 6
`path` (`effipy.tariffs.Tariffs` attribute), 12
`path` (`effipy.usagepoints.UsagePoints` attribute), 14

R

`reading()` (`effipy.meters.Meters` method), 15
`readings()` (`effipy.meters.Meters` method), 15
`require()` (`effipy.contracts.Contracts` method), 10
`require()` (`effipy.customers.Customers` method), 9
`require()` (`effipy.invoices.Invoices` method), 11
`require()` (`effipy.meters.Meters` method), 15
`require()` (`effipy.resource.EffiPeopleResource` method), 6
`require()` (`effipy.service.EffiPeople` method), 7
`require()` (`effipy.tariffs.Tariffs` method), 12
`require()` (`effipy.usagepoints.UsagePoints` method), 14
`require_collection()` (`effipy.contracts.Contracts` method), 10
`require_collection()` (`effipy.customers.Customers` method), 9
`require_collection()` (`effipy.invoices.Invoices` method), 11
`require_collection()` (`effipy.meters.Meters` method), 15
`require_collection()` (`effipy.resource.EffiPeopleResource` method), 6
`require_collection()` (`effipy.tariffs.Tariffs` method), 12
`require_collection()` (`effipy.usagepoints.UsagePoints` method), 14
`require_item()` (`effipy.contracts.Contracts` method), 10
`require_item()` (`effipy.customers.Customers` method), 9
`require_item()` (`effipy.invoices.Invoices` method), 11
`require_item()` (`effipy.meters.Meters` method), 15
`require_item()` (`effipy.resource.EffiPeopleResource` method), 6
`require_item()` (`effipy.tariffs.Tariffs` method), 12
`require_item()` (`effipy.usagepoints.UsagePoints` method), 14

T

`tariff()` (`effipy.service.EffiPeople` method), 7
Tariffs (class in `effipy.tariffs`), 11
`tariffs()` (`effipy.service.EffiPeople` method), 7
`Tariffs.create()` (built-in function), 11
`Tariffs.get()` (built-in function), 12

U

`update()` (`effipy.contracts.Contracts` method), 10
`update()` (`effipy.customers.Customers` method), 9
`update()` (`effipy.invoices.Invoices` method), 11
`update()` (`effipy.meters.Meters` method), 15
`update()` (`effipy.resource.EffiPeopleResource` method), 6
`update()` (`effipy.tariffs.Tariffs` method), 12
`update()` (`effipy.usagepoints.UsagePoints` method), 14
`usagepoint()` (`effipy.service.EffiPeople` method), 7
UsagePoints (class in `effipy.usagepoints`), 13
`usagepoints()` (`effipy.service.EffiPeople` method), 7
`UsagePoints.create()` (built-in function), 13
`UsagePoints.get()` (built-in function), 13
`use_json()` (`effipy.service.EffiPeople` method), 7

V

`version()` (`effipy.contracts.Contracts` method), [10](#)
`version()` (`effipy.tariffs.Tariffs` method), [13](#)
`versions()` (`effipy.contracts.Contracts` method), [10](#)
`versions()` (`effipy.tariffs.Tariffs` method), [13](#)

W

`wrap_object()` (`effipy.contracts.Contracts` method), [10](#)
`wrap_object()` (`effipy.customers.Customers` method), [9](#)
`wrap_object()` (`effipy.invoices.Invoices` method), [11](#)
`wrap_object()` (`effipy.meters.Meters` method), [15](#)
`wrap_object()` (`effipy.resource.EffiPeopleResource` method), [6](#)
`wrap_object()` (`effipy.tariffs.Tariffs` method), [13](#)
`wrap_object()` (`effipy.usagepoints.UsagePoints` method), [14](#)