
Prometeo Python

Release 1.2.2

Prometeo

Mar 11, 2022

CONTENTS

1	Installation	3
2	Quick Start	5
3	Usage Guide	7
4	Reference	15
5	Indices and tables	37
	Python Module Index	39
	Index	41

The official python package for the [Prometeo API](#)

INSTALLATION

Install from PyPI using `pip`:

```
$ pip install prometeo
```


QUICK START

Go to your [dashboard](#), there you'll find your API key. Use it to instantiate the client:

```
from prometeo import Client  
  
client = Client('<YOUR_API_KEY>', environment='testing')
```

The environment argument is either testing for the sandbox or production for the production environment.

Note: [Contact us](#) to get an api key!

USAGE GUIDE

Guides for all the available services:

3.1 Banking API

3.1.1 Initialize the client

To initialize the client you'll need to provide your api key and the environment (either `testing` or `production`)

```
from prometeo import Client

client = Client('<YOUR_API_KEY>', environment='testing')
```

3.1.2 Log in

```
session = client.banking.login(
    provider='test',
    username='12345',
    password='gfdsa',
)
```

To get a list of available provider codes, use `get_providers()`

Beuase some banks require extra fields to log in, these must be provided inside `kwargs` as a dictionary. To see if a provider requires extra auth fields to log in, use `get_provider_detail()`

Example with extra auth fields:

```
session = client.banking.login(
    provider='bbva_mx',
    username='12345',
    password='gfdsa',
    otp=11223
)
```

The following is a table with all the extra auth fields and their respective providers.

Table 1: Extra Login fields

Field	Provider	Required	Description
otp	bbva_mx	True	One-time password
personal_question	banco_general, inter-matico	True	Personal question
token	banorte	True	Provider authorization token
company_code	pe_bbva_netcash	True	Company code
type	davivienda	False	Options <ul style="list-style-type: none"> cedula_ciudadania: citizenship card cedula_extranjeria: foreigner Identity card nit NIT tarjeta_identidad: identity card pasaporte: passport tarjeta_seguro_social_e: foreign Social Security card sociedad_extranjera_sin: foreign company without NIT in Colombia fideicomiso: fideicomiso nit_menores: Minor NIT rif_venezuela: RIF Venezuela nit_extranjeria: Foreigners NIT nit_persona_natural: Natural Person NIT registro_civil_nacimiento: birth certificate
8			Chapter 3. Usage Guide <ul style="list-style-type: none"> nit_desasociado: disassociated NIT

3.1.3 Select client

In some banks a user can have access to more than one profile (called client), in those cases, the session status after login will be `select_client`. It is then necessary to select the client, for that, first list the available clients and pass one to `session.select_client`

```
if session.get_status() == 'select_client':
    clients = session.get_clients()
    session.select_client(clients[0])
    assert session.status == 'logged_in'
```

If the bank doesn't use multiple clients, calling `get_clients` will return an empty list.

3.1.4 Handling security questions and OTPs

In cases where the bank requires additional steps to login, such as answering a personal security question or using a 2FA device like an OTP (One Time Password), the status of the session will be set as `interaction_required`, which can be handled like this:

```
session = client.login(provider='test', username='user', password='pass')
if session.get_status() == 'interaction_required':
    # necessary context, like the security question to answer.
    print(session.get_interactive_context())
    session.finish_login(
        provider='test',
        username='user',
        password='pass',
        answer='1234',
    )
```

3.1.5 Restoring a session

In some cases it may be useful to serialize the session to be used later or to transfer to another process, like in a task queue job. For this use `get_session_key()` and `get_session()`:

```
session_key = session.get_session_key()

# save session_key somewhere...

restored_session = client.banking.get_session(session_key)
```

3.1.6 Listing accounts and movements

```
from datetime import datetime

accounts = session.get_accounts()
for account in accounts:
    movements = account.get_movements(
        datetime(2019, 2, 1), datetime(2019, 15, 1)
    )
```

For more detailed information, refer to the docs for `get_accounts()` and `get_movements()`

3.1.7 Listing credit cards and their movements

Credit cards can have movements in more than one currency, so it's necessary to specify it when listing movements.

```
from datetime import datetime

cards = session.get_credit_cards()
for card in cards:
    movements = card.get_movements(
        'USD', datetime(2019, 2, 1), datetime(2019, 15, 1)
    )
```

3.1.8 Listing available banks

We recommend that the list of available banks be stored on a database and updated weekly.

```
providers = client.banking.get_providers()
```

3.1.9 Preprocess transfer

```
preprocess = session.preprocess_transfer(
    origin_account='002206345988',
    destination_institution='0',
    destination_account='001002363321',
    currency='UYU',
    amount='1.3',
    concept='transfer description',
    destination_owner_name='John Doe',
    branch='62',
)

print(preprocess)
```

3.1.10 Confirm transfer

```
confirmation = session.confirm_transfer(
    request_id='0b7d6b32d1be4c11bde21e7ddc08cc36',
    authorization_type='cardCode',
    authorization_data='1, 2, 3',
)

print(confirmation)
```

3.1.11 List transfer institutions

```
institutions_list = session.list_transfer_institutions()
for institution in institutions_list:
    print(institution)
```

3.2 CURP API

3.2.1 Checking the existence of a curp

```
from prometeo import Client
from prometeo.curp import exceptions

client = Client('<YOUR_API_KEY>', environment='testing')
try:
    result = client.curp.query('ABCD12345EFGH')
except exceptions.CurpError as e:
    print("CURP does not exist:", e.message)
```

3.2.2 Looking for a CURP by personal info

```
from datetime import datetime
from prometeo.curp import exceptions, Gender, State

curp = 'ABCD12345EFGH'
state = State.SINALOA
birthdate = datetime(1988, 3, 4)
name = 'JOHN'
first_surname = 'DOE'
last_surname = 'PONCE'
gender = Gender.MALE
try:
    result = client.curp.reverse_query(
        state, birthdate, name, first_surname, last_surname, gender
    )
except exceptions.CurpError as e:
    print("CURP does not exist:", e.message)
```

Check the [reference](#) for a list of possible values for State

3.3 DIAN API

3.3.1 Log in

Initialize the client:

```
from prometeo import Client
from prometeo.dian import DocumentType

client = Client('<YOUR_API_KEY>', environment='testing')
```

Supply the NIT to log in as a company

```
session = client.dian.login(
    nit='098765',
    document_type=DocumentType.CEDULA_CIUADANIA,
    document='12345',
    password='test_password',
)
```

Or omit it to log in as a person:

```
session = client.dian.login(
    document_type=DocumentType.CEDULA_CIUADANIA,
    document='12345',
    password='test_password',
)
```

Check the [reference](#) for a list of possible values for `DocumentType`

3.3.2 Restoring a session

In some cases it may be useful to serialize the session to be used later or to transfer to another process, like in a task queue job. For this use `get_session_key()` and `get_session()`:

```
session_key = session.get_session_key()

# save session_key somewhere...

restored_session = client.dian.get_session(session_key)
```

3.3.3 Download the forms

To download the original forms in pdf format, use the `pdf` property of the object returned by the methods:

```
info = session.get_company_info()
pdf_content = info.pdf.get_content()
# write the contents to a file:
with open("company-info.pdf", "wb") as f:
    f.write(pdf_content)
```

3.3.4 Getting the data

Company info (form 001):

```
session.get_company_info()
```

Balances:

```
session.get_balances()
```

Rent declaration:

```
session.get_rent_declaration(2019)
```


VAT declaration:

```
from prometeo.dian import Periodicity, QuartlerlyPeriod

session.get_vat_declaration(2019, Periodicity.QUARTERLY, QuartlerlyPeriod.JANUARY_
    ↳APRIL)
```

Numeration:

```
from datetime import datetime
from prometeo.dian import NumerationType

session.get_numeration(
    NumerationType.Authorization,
    datetime(2019, 1, 1),
    datetime(2019, 5, 1)
)
```

Retentions:

```
from prometeo.dian import MonthlyPeriod

session.get_retentions(2019, MonthlyPeriod.NOVEMBER)
```

3.4 SAT API

3.4.1 Log in

```
from prometeo.sat import LoginScope

session = client.sat.login(
    rfc='ABCD1234EFGH',
    password='password',
    scope=LoginScope.CFDI,
)
```

3.4.2 Restoring a session

In some cases it may be useful to serialize the session to be used later or to transfer to another process, like in a task queue job. For this use `get_session_key()` and `get_session()`:

```
session_key = session.get_session_key()

# save session_key somewhere...

restored_session = client.sat.get_session(session_key)
```

3.4.3 Listing bills

To list emitted bills, supply a date range and the *BillStatus*

```
from prometeo.sat import BillStatus

emitted_bills = session.get_emitted_bills(
    date_start=datetime(2020, 1, 1),
    date_end=datetime(2020, 2, 1),
    status=BillStatus.ANY,
)
```

And for received bills, supply the year, month and the *BillStatus*

```
received_bills = session.get_received_bills(
    year=2020,
    month=1,
    status=BillStatus.ANY,
)
```

Check the documentation for *CFDIBill* to see a list of all fields available for a bill.

3.4.4 Downlading bills

```
import time
from prometeo.sat import BillStatus

download_requests = session.download_emitted_bills(
    date_start=datetime(2020, 1, 1),
    date_end=datetime(2020, 2, 1),
    status=BillStatus.ANY,
)
for request in download_requests:
    while not request.is_ready():
        time.sleep(5)
    download = request.get_download()
    content = download.get_file().read()
```

3.4.5 Download acknowledgements

```
from prometeo.sat import Motive, DocumentType, Status, SendType

acks = session.get_acknowledgement(
    year=2020,
    month_start=1,
    month_end=2,
    motive=Motive.ALL,
    document_type=DocumentType.ALL,
    status=Status.ALL,
    send_type=SendType.ALL,
)
for ack in acks:
    download = ack.download().get_file()
```

REFERENCE

4.1 Client Reference

A complete documentation of the prometeo module.

4.1.1 prometeo.banking

API Client

class prometeo.banking.client.**Account** (*client, session_key, account_data*)
A bank account, returned by *get_accounts()*

get_movements (*date_start, date_end*)
List an account's movements for a range of dates.

Parameters

- **date_start** (datetime) – Start of the date range for movements.
- **date_end** (datetime) – End of the date range for movements.

Return type List of *Movement*

class prometeo.banking.client.**BankingAPIClient** (*api_key, environment*)
API Client for banking api

confirm_transfer (*session_key, request_id, authorization_type, authorization_data*)
Confirm transfer.

Parameters

- **session_key** (*str*) – The session key.
- **request_id** (*str*) – Id of the request returned by the endpoint of *preprocess_transfer()*
- **authorization_type** (*str*) –
 - cardCode Coordinates card
 - pin Account personal pin
 - otp One-time generated pin, sent by sms, email, hard token or soft token.
 - otp-api Hard token or soft token device, digitized by Prometeo
- **authorization_data** (*str*) – Verification value (pin number, coordinates card response , etc) if there are several values, they must be separated by commas.

Return type *ConfirmTransfer*

get_provider_detail (*provider_code*)

Get more detailed information about a bank.

Parameters **provider_code** (*str*) – Name of the provider, as returned in *get_providers()*

Return type *ProviderDetail*

get_providers ()

List all available banks.

Return type *Provider*

list_transfer_institutions (*session_key*)

List transfer institutions.

Parameters **session_key** (*str*) – The session key.

Return type *TransferInstitution*

login (*provider, username, password, **kwargs*)

Start log in process with the provider

Parameters

- **provider** (*str*) – Name of the provider, use *get_providers()* to get a list of available providers
- **username** (*str*) – Username used to log in to the banking app or web
- **password** (*str*) – User's password
- **kwargs** (*dict*) – Extra login fields for providers that require it, use *get_provider_detail()* to get a list of all the auth login fields

Return type *Session*

logout (*session_key*)

Logs the user out and invalidates its session.

Parameters **session_key** (*str*) – The session key.

on_response (*data*)

Called after every 200 response

preprocess_transfer (*session_key, origin_account, destination_institution, destination_account, currency, amount, concept, destination_owner_name, branch*)

Preprocess transfer.

Parameters

- **session_key** (*str*) – The session key.
- **origin_account** (*str*) – Account number from where it is transferred
- **destination_institution** (*str*) – Id of the institution where it is transferred. As provided by *get_providers()*
- **destination_account** (*str*) – Account number where it is transferred
- **currency** (*str*) – Account currency in format ISO 4217
- **amount** (*str*) – Amount to transfer
- **concept** (*str*) – Concept or description of the transfer

- **destination_owner_name** (*str*) – Name of the holder of the destination account (empty if not applicable)
- **branch** (*str*) – Branch number of the destination account (empty if not applicable)

Return type *PreprocessTransfer*

session_class

alias of *Session*

class prometeo.banking.client.**CreditCard**(*client, session_key, card_data*)

A credit card, returned by *get_credit_cards()*

get_movements (*currency_code, date_start, date_end*)

List credit card's movements for a range of dates.

Parameters

- **date_start** (datetime) – Start of the date range for movements.
- **date_end** (datetime) – End of the date range for movements.

Return type List of *Movement*

class prometeo.banking.client.**Session**(*client, status, session_key, context=None, field=None*)

Encapsulates the user's session, returned by *login()*

confirm_transfer (*request_id, authorization_type, authorization_data*)

Confirm transfer.

Parameters

- **request_id** (*str*) – Id of the request returned by the endpoint of *preprocess_transfer()*
- **authorization_type** (*str*) –
 - cardCode Coordinates card
 - pin Account personal pin
 - otp One-time generated pin, sent by sms, email, hard token or soft token.
 - otp-api Hard token or soft token device, digitized by Promete
- **authorization_data** (*str*) – Verification value (pin number, coordinates card response , etc) if there are several values, they must be separated by commas.

Return type *ConfirmTransfer*

finish_login (*provider, username, password, answer*)

Answer the security challenge, like an OTP or personal question.

Parameters

- **provider** (*str*) – The provider used to login
- **username** (*str*) – The username used to login
- **password** (*str*) – The password used to login
- **answer** (*str*) – The answer to the login challenge

get_accounts ()

List all the user's accounts

Return type List of *Account*

get_clients()

List the user's clients. Returns an empty list if the bank doesn't uses clients.

Return type List of *Client*

get_credit_cards()

List all the user's credit cards

Return type List of *CreditCard*

get_interactive_context()

Necessary information to answer the login challenge, like a security question.

Return type str

list_transfer_institutions()

List transfer institutions.

Return type *TransferInstitution*

logout()

Logs the user out and invalidates its session.

preprocess_transfer(*origin_account*, *destination_institution*, *destination_account*, *currency*,
amount, *concept*, *destination_owner_name*, *branch*)

Preprocess transfer.

Parameters

- **origin_account** (*str*) – Account number from where it is transferred
- **destination_institution** (*str*) – Id of the institution where it is transferred. As provided by *get_providers()*
- **destination_account** (*str*) – Account number where it is transferred
- **currency** (*str*) – Account currency in format ISO 4217
- **amount** (*str*) – Amount to transfer
- **concept** (*str*) – Concept or description of the transfer
- **destination_owner_name** (*str*) – Name of the holder of the destination account (empty if not applicable)
- **branch** (*str*) – Branch number of the destination account (empty if not applicable)

Return type *PreprocessTransfer*

select_client(*client*)

Selects the client to use for this session.

Parameters **client_id** (*str*) – The id of the client, obtained from listing the clients

Models

```
class prometeo.banking.models.Account (id, name, number, branch, currency, balance)
```

```
    property balance
        Alias for field number 5
```

```
    property branch
        Alias for field number 3
```

```
    property currency
        Alias for field number 4
```

```
    property id
        Alias for field number 0
```

```
    property name
        Alias for field number 1
```

```
    property number
        Alias for field number 2
```

```
class prometeo.banking.models.Client (id, name)
```

```
    property id
        Alias for field number 0
```

```
    property name
        Alias for field number 1
```

```
class prometeo.banking.models.ConfirmTransfer (message, success)
```

```
    property message
        Alias for field number 0
```

```
    property success
        Alias for field number 1
```

```
class prometeo.banking.models.CreditCard (id, name, number, close_date, due_date, balance_local, balance_dollar)
```

```
    property balance_dollar
        Alias for field number 6
```

```
    property balance_local
        Alias for field number 5
```

```
    property close_date
        Alias for field number 3
```

```
    property due_date
        Alias for field number 4
```

```
    property id
        Alias for field number 0
```

```
    property name
        Alias for field number 1
```

```
    property number
        Alias for field number 2
```

```
class prometeo.banking.models.Movement (id, reference, date, detail, debit, credit, extra_data)
```

```
    property credit  
        Alias for field number 5
```

```
    property date  
        Alias for field number 2
```

```
    property debit  
        Alias for field number 4
```

```
    property detail  
        Alias for field number 3
```

```
    property extra_data  
        Alias for field number 6
```

```
    property id  
        Alias for field number 0
```

```
    property reference  
        Alias for field number 1
```

```
class prometeo.banking.models.PreprocessTransfer (approved, authorization_devices,  
                                                message, request_id)
```

```
    property approved  
        Alias for field number 0
```

```
    property authorization_devices  
        Alias for field number 1
```

```
    property message  
        Alias for field number 2
```

```
    property request_id  
        Alias for field number 3
```

```
class prometeo.banking.models.Provider (code, country, name)
```

```
    property code  
        Alias for field number 0
```

```
    property country  
        Alias for field number 1
```

```
    property name  
        Alias for field number 2
```

```
class prometeo.banking.models.ProviderDetail (country, name, auth_fields)
```

```
    property auth_fields  
        Alias for field number 2
```

```
    property country  
        Alias for field number 0
```

```
    property name  
        Alias for field number 1
```



```
class prometeo.banking.models.TransferInstitution(id, name)

    property id
        Alias for field number 0

    property name
        Alias for field number 1
```

4.1.2 base

Base Client

```
class prometeo.base_client.BaseClient(api_key, environment)
    Base client class to make api calls

    call_api(method, url, *args, **kwargs)
        Calls an API endpoint, using the configured api key and environment.

        Parameters
            • method (str) – The HTTP method to use (GET, POST, etc)
            • url (str) – The url to call (without the environment's domain)

        Return type JSON data as a python object.

    get_session(session_key)
        Restore a session from its session key

        Parameters session_key (str) – The session key

        Return type Session

    on_response(response_data)
        Called after every 200 response

class prometeo.base_client.Download(client, url)
    Represents a downloadable file, like an xml bill or pdf document

    get_file()
        Downloads the file and returns its contents.

        Return type bytes
```

Base Session

```
class prometeo.base_session.BaseSession(client, status, session_key)
    Base class that handles calling the API endpoints that use session keys.

    get_session_key()
        Returns this session's session key

        Return type str

    get_status()
        Returns this session's status

        Return type str
```

4.1.3 prometeo.curp

API Client

class prometeo.curp.client.CurpAPIClient (*api_key, environment*)
API Client for CURP queries

query (*curp*)

Find the personal data associated with a CURP

Parameters **curp** (*str*) – The CURP of the person to query

Return type *QueryResult*

reverse_query (*state, birthdate, name, first_surname, last_surname, gender*)

Search for a person by their personal information.

Parameters

- **state** (*State*) – The state where the person is registered
- **birthdate** (*datetime*) – The person's birthdate
- **name** (*str*) – The person's name
- **first_surname** (*str*) – The person's first surname
- **last_surname** (*str*) – The person's last surname
- **gender** (*Gender*) – The person's gender

Return type *QueryResult*

Enums

class prometeo.curp.client.Gender
A person's gender

FEMALE = 'M'

MALE = 'H'

class prometeo.curp.client.State
The state a person is registered

AGUASCALIENTES = 'AS'

BAJA_CALIFORNIA = 'BC'

BAJA_CALIFORNIA_SUR = 'BS'

CAMPECHE = 'CC'

CHIAPAS = 'CS'

CHIHUAHUA = 'CH'

CIUDAD_DE_MEXICO = 'DF'

COAHUILA = 'CL'

COLIMA = 'CM'

DURANGO = 'DG'

ESTADO_DE_MEXICO = 'MC'

```

GUANAJUATO = 'GT'
GUERRERO = 'GR'
HIDALGO = 'HG'
JALISCO = 'JC'
MICHOACAN = 'MN'
MORELOS = 'MS'
NACIDO_EN_EL_EXTRANJERO = 'NE'
NAYARIT = 'NT'
NUEVO_LEON = 'NL'
OAXACA = 'OC'
PUEBLA = 'PL'
QUERETARO = 'QT'
QUINTANA_ROO = 'QR'
SAN_LUIS_POTOSI = 'SP'
SINALOA = 'SL'
SONORA = 'SR'
TABASCO = 'TC'
TAMAULIPAS = 'TS'
TLAXCALA = 'TL'
VERACRUZ = 'VZ'
YUCATAN = 'YN'
ZACATECA = 'ZS'

```

Models

```

class prometeo.curp.models.DocumentData(entidad_registro, clave_entidad_registro, libro,
                                         clave_municipio_registro, municipio_registro,
                                         anio_reg, tomo, num_acta, foja)

    property anio_reg
        Alias for field number 5

    property clave_entidad_registro
        Alias for field number 1

    property clave_municipio_registro
        Alias for field number 3

    property entidad_registro
        Alias for field number 0

    property foja
        Alias for field number 8

```

property libro
Alias for field number 2

property municipio_registro
Alias for field number 4

property num_acta
Alias for field number 7

property tomo
Alias for field number 6

`prometeo.curp.models.PersonalData`
alias of `prometeo.curp.models.DocumentData`

class `prometeo.curp.models.QueryResult` (*document_data, personal_data, pdf_url, pdf*)

property document_data
Alias for field number 0

property pdf
Alias for field number 3

property pdf_url
Alias for field number 2

property personal_data
Alias for field number 1

4.1.4 prometeo.dian

API Client

class `prometeo.dian.client.DianAPIClient` (*api_key, environment*)
API Client for DIAN API

login (*document_type, document, password, nit=None*)
Log in to DIAN

Parameters

- **document_type** (*DocumentType*) – The type of document for the document argument.
- **document** (*str*) – The document of the person
- **password** (*str*) – The password used to log in
- **nit** (*str*) – The NIT of the company (used only when logging in as a company)

Return type *Session*

session_class
alias of *Session*

class `prometeo.dian.client.Session` (*client, status, session_key*)

get_balances ()
Get balances.

Return type list of *Balance*

get_company_info()

Get information about the company or person (form 001).

Return type *CompanyInfo*

get_numeration(*type, date_start, date_end*)

Get bill numeration (form 1876)

Parameters

- **type** (*NumerationType*) – The type of numeration request to get
- **date_start** (datetime) – Start date to filter
- **date_end** (datetime) – End date to filter

Return type List of *Numeration*

get_rent_declaration(*year*)

Get rent declaration information (form 110 for companies and 210 for persons)

Parameters **year** (*int*) – Year of the declaration

Return type *RentDeclaration*

get_retentions(*year, period*)

Get retentions information (form 350)

Parameters

- **year** (*int*) – Year of the retention
- **period** (*int*) – Period of the retention

Return type List of *Retentions*

get_vat_declaration(*year, periodicity, period*)

Get VAT declaration information (form 300)

Parameters

- **year** (*int*) – Year of the declaration
- **periodicity** (*Periodicity*) – Periodicity, either quarterly or bimonthly
- **period** (*int*) – Period of the declaration

Return type *VATDeclaration*

Enums

class prometeo.dian.client.NumerationType

An enumeration.

Authorization = 'authorization'

Habilitation = 'habilitation'

Inhabilitation = 'inhabilitation'

class prometeo.dian.client.Periodicity

An enumeration.

BIMONTHLY = 'b'

QUARTERLY = 'q'

```
class prometeo.dian.client.QuarterlyPeriod
    An enumeration.

    JANUARY_APRIL = 1

    MAY_AUGUST = 2

    SEPTEMBER_DECEMBER = 3

class prometeo.dian.client.BimonthlyPeriod
    An enumeration.

    JANUARY_FEBRUARY = 1

    JULY_AUGUST = 4

    MARCH_APRIL = 2

    MAY_JUNE = 3

    NOVEMBER_DECEMBER = 6

    SEPTEMBER_OCTOBER = 5

class prometeo.dian.client.MonthlyPeriod
    An enumeration.

    APRIL = 4

    AUGUST = 8

    DECEMBER = 12

    FEBRUARY = 2

    JANUARY = 1

    JULY = 7

    JUNE = 6

    MARCH = 3

    MAY = 5

    NOVEMBER = 11

    OCTOBER = 10

    SEPTEMBER = 9

class prometeo.dian.client.DocumentType
    Document types used for DianAPIClient.login()

    CARNE_DIPLOMATICO = '46'

    CEDULA_CIUDADANIA = '13'

    CEDULA_EXTRANJERIA = '22'

    CERTIFICADO_REGISTRADURIA = '14'

    DOCUMENTO_IDENTIFICACION_EXTRANJERO = '42'

    DOCUMENTO_IDENTIFICACION_EXTRANJERO_PERSONA_JURIDICA = '44'

    PASAPORTE = '41'

    SIN_IDENTIFICACION_EXTRANJERO = '43'
```

```
TARJETA_EXTRANJERIA = '21'
```

```
TARJETA_IDENTIDAD = '12'
```

Models

```
class prometeo.dian.models.Accountant (document, start_date, name, professional_card)
```

```
    property document
```

```
        Alias for field number 0
```

```
    property name
```

```
        Alias for field number 2
```

```
    property professional_card
```

```
        Alias for field number 3
```

```
    property start_date
```

```
        Alias for field number 1
```

```
class prometeo.dian.models.Balance (type, amount, value)
```

```
    property amount
```

```
        Alias for field number 1
```

```
    property type
```

```
        Alias for field number 0
```

```
    property value
```

```
        Alias for field number 2
```

```
class prometeo.dian.models.CapitalComposition (national, national_private, national_public, foreign, foreign_private, foreign_public)
```

```
    property foreign
```

```
        Alias for field number 3
```

```
    property foreign_private
```

```
        Alias for field number 4
```

```
    property foreign_public
```

```
        Alias for field number 5
```

```
    property national
```

```
        Alias for field number 0
```

```
    property national_private
```

```
        Alias for field number 1
```

```
    property national_public
```

```
        Alias for field number 2
```

```
class prometeo.dian.models.CompanyInfo (accountant, capital_composition, reason, pdf_url, pdf, location, name, constitution_date, representation, members)
```

```
    property accountant
```

```
        Alias for field number 0
```

property capital_composition

Alias for field number 1

property constitution_date

Alias for field number 7

property location

Alias for field number 5

property members

Alias for field number 9

property name

Alias for field number 6

property pdf

Alias for field number 4

property pdf_url

Alias for field number 3

property reason

Alias for field number 2

property representation

Alias for field number 8

class prometeo.dian.models.**Field**(*name, number, value*)

property name

Alias for field number 0

property number

Alias for field number 1

property value

Alias for field number 2

class prometeo.dian.models.**Location**(*country, city, phone1, phone2, department, address, email*)

property address

Alias for field number 5

property city

Alias for field number 1

property country

Alias for field number 0

property department

Alias for field number 4

property email

Alias for field number 6

property phone1

Alias for field number 2

property phone2

Alias for field number 3


```
class prometeo.dian.models.Member (document_type, document, nationality, name, start_date)
```

```
    property document
```

```
        Alias for field number 1
```

```
    property document_type
```

```
        Alias for field number 0
```

```
    property name
```

```
        Alias for field number 3
```

```
    property nationality
```

```
        Alias for field number 2
```

```
    property start_date
```

```
        Alias for field number 4
```

```
class prometeo.dian.models.Name (first_surname, second_surname, first_name, other_names)
```

```
    property first_name
```

```
        Alias for field number 2
```

```
    property first_surname
```

```
        Alias for field number 0
```

```
    property other_names
```

```
        Alias for field number 3
```

```
    property second_surname
```

```
        Alias for field number 1
```

```
class prometeo.dian.models.Numeration (nit, dv, name, reason, address, country, department,  
                                         municipality, ranges, pdf_url, pdf_available)
```

```
    property address
```

```
        Alias for field number 4
```

```
    property country
```

```
        Alias for field number 5
```

```
    property department
```

```
        Alias for field number 6
```

```
    property dv
```

```
        Alias for field number 1
```

```
    property municipality
```

```
        Alias for field number 7
```

```
    property name
```

```
        Alias for field number 2
```

```
    property nit
```

```
        Alias for field number 0
```

```
    property pdf_available
```

```
        Alias for field number 10
```

```
    property pdf_url
```

```
        Alias for field number 9
```

property ranges

Alias for field number 8

property reason

Alias for field number 3

`prometeo.dian.models.NumerationRange`

alias of `prometeo.dian.models.Range`

class `prometeo.dian.models.RentDeclaration` (*pdf_url, pdf, fields, year, form_number, nit, dv, name, reason, direction_code, economic_activity, correction_code, previous_form*)

property correction_code

Alias for field number 11

property direction_code

Alias for field number 9

property dv

Alias for field number 6

property economic_activity

Alias for field number 10

property fields

Alias for field number 2

property form_number

Alias for field number 4

property name

Alias for field number 7

property nit

Alias for field number 5

property pdf

Alias for field number 1

property pdf_url

Alias for field number 0

property previous_form

Alias for field number 12

property reason

Alias for field number 8

property year

Alias for field number 3

class `prometeo.dian.models.Representative` (*document, document_type, name, representation_type, start_date*)

property document

Alias for field number 0

property document_type

Alias for field number 1

property name
Alias for field number 2

property representation_type
Alias for field number 3

property start_date
Alias for field number 4

class prometeo.dian.models.**Retentions** (*pdf_url, pdf, fields, year, period, form_number, nit, reason, direction_code*)

property direction_code
Alias for field number 8

property fields
Alias for field number 2

property form_number
Alias for field number 5

property nit
Alias for field number 6

property pdf
Alias for field number 1

property pdf_url
Alias for field number 0

property period
Alias for field number 4

property reason
Alias for field number 7

property year
Alias for field number 3

class prometeo.dian.models.**VATDeclaration** (*pdf_url, pdf, fields, year, period, form_number, nit, dv, name, reason, direction_code, correction_code, previous_form*)

property correction_code
Alias for field number 11

property direction_code
Alias for field number 10

property dv
Alias for field number 7

property fields
Alias for field number 2

property form_number
Alias for field number 5

property name
Alias for field number 8

property nit
Alias for field number 6

property pdf
Alias for field number 1

property pdf_url
Alias for field number 0

property period
Alias for field number 4

property previous_form
Alias for field number 12

property reason
Alias for field number 9

property year
Alias for field number 3

4.1.5 prometeo.sat

API Client

class `prometeo.sat.client.SatAPIClient` (*api_key, environment*)
API Client for SAT API

login (*rfc, password, scope*)
Log in to SAT

Parameters

- **rfc** (*str*) – RFC of the person to log in
- **password** (*str*) – Password used to login, also known as CIEC
- **scope** (*LoginScope*) – Depending on the type of information to query, use `LoginScope.CFDI` to download bill xmls or `LoginScope.SIAT` to download acknowledgements.

Return type *Session*

session_class
alias of *Session*

class `prometeo.sat.client.Session` (*client, status, session_key*)

download_emitted_bills (*date_start, date_end, status*)
Creates a request to download all the emitted bills in a range of dates.

Parameters

- **date_start** (*datetime*) – Start date to filter
- **date_end** (*datetime*) – End date to filter
- **status** (*BillStatus*) – Status of the bills

Return type List of *DownloadRequest*

download_received_bills (*year, month, status*)
Creates a request to download all the received bills in a range of dates.

Parameters

- **date_start** (datetime) – Start date to filter
- **date_end** (datetime) – End date to filter
- **status** (*BillStatus*) – Status of the bills

Return type List of *DownloadRequest*

get_acknowledgements (*year, month_start, month_end, motive, document_type, status, send_type*)
Gets a list of acknowledgements for a range of dates

Parameters

- **year** (*int*) – The year of the acknowledgements
- **month_start** (*int*) – Start month to filter
- **month_end** (*int*) – End month to filter
- **motive** (*Motive*) – Motive
- **document_type** (*DocumentType*) – Document type
- **status** (*Status*) – Status
- **send_type** (*SendType*) – Send type

Return type List of *AcknowledgementResult*

get_downloads ()
Gets a list of available downloads

Return type List of *DownloadRequest*

get_emitted_bills (*date_start, date_end, status*)
List all emitted bills in a range of dates.

Parameters

- **date_start** (datetime) – Start date to filter
- **date_end** (datetime) – End date to filter
- **status** (*BillStatus*) – Status of the bills

Return type List of *CFDIBill*

get_received_bills (*year, month, status*)
List all received bills in a range of dates.

Parameters

- **year** (*int*) – Year of the received bills
- **month** (*int*) – Month of the received bills
- **status** (*BillStatus*) – Status of the bills

Return type List of *CFDIBill*

logout ()
Logs out of SAT. You won't be able to use this session after logout.

class prometeo.sat.client.**AcknowledgementResult** (*client, session_key, data*)
Info on an acknowledgement, returned by *Session.get_acknowledgements* ()

get_download ()
Download the acknowledgement data

Return type Download

class prometeo.sat.client.DownloadRequest (*client, session_key, request_id*)
A request for bulk downloading of bills.

get_download()
Download the generated zip file with all the xmls

Return type Download

is_ready()
Check if the request is ready to download.

Return type bool

Enums

class prometeo.sat.client.LoginScope
An enumeration.

CFDI = 'cfdi'

SIAT = 'siat'

class prometeo.sat.client.BillStatus
An enumeration.

ANY = 'any'

CANCELLED = 'cancelled'

VALID = 'valid'

class prometeo.sat.client.Motive
An enumeration.

AF = 'af'

ALL = 'all'

CO = 'co'

DE = 'de'

FC = 'fc'

MONTHLY = 'monthly'

class prometeo.sat.client.DocumentType
An enumeration.

ALL = 'all'

B = 'b'

CT = 'ct'

PL = 'pl'

XC = 'xc'

XF = 'xf'

class prometeo.sat.client.Status
An enumeration.

ACCEPTED = 'accepted'

```

    ALL = 'all'
    RECEIVED = 'received'
    REJECTED = 'rejected'
class prometeo.sat.client.SendType
    An enumeration.
    ALL = 'all'
    C = 'c'
    N = 'n'

```

Models

```

class prometeo.sat.models.AcknowledgementResult(id, period, motive, document_type,
                                                send_type, file_name, reception_date,
                                                status)

    property document_type
        Alias for field number 3
    property file_name
        Alias for field number 5
    property id
        Alias for field number 0
    property motive
        Alias for field number 2
    property period
        Alias for field number 1
    property reception_date
        Alias for field number 6
    property send_type
        Alias for field number 4
    property status
        Alias for field number 7
class prometeo.sat.models.CFDIBill(id, emitter_rfc, emitter_reason, receiver_rfc, re-
                                   ceiver_reason, emitted_date, certification_date, certi-
                                   fication_pac, total_value, effect, status)

    property certification_date
        Alias for field number 6
    property certification_pac
        Alias for field number 7
    property effect
        Alias for field number 9
    property emitted_date
        Alias for field number 5
    property emitter_reason
        Alias for field number 2

```

property emitter_rfc

Alias for field number 1

property id

Alias for field number 0

property receiver_reason

Alias for field number 4

property receiver_rfc

Alias for field number 3

property status

Alias for field number 10

property total_value

Alias for field number 8

class prometeo.sat.models.CFDIDownloadItem(*request_id, type, count*)

property count

Alias for field number 2

property request_id

Alias for field number 0

property type

Alias for field number 1

class prometeo.sat.models.DownloadFile(*download_url*)

property download_url

Alias for field number 0

class prometeo.sat.models.DownloadRequest(*request_id*)

property request_id

Alias for field number 0

class prometeo.sat.models.PdfFile(*pdf_url*)

property pdf_url

Alias for field number 0

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

p

- `prometeo.banking`, [15](#)
- `prometeo.banking.client`, [15](#)
- `prometeo.banking.models`, [19](#)
- `prometeo.base_client`, [21](#)
- `prometeo.base_session`, [21](#)
- `prometeo.curp`, [22](#)
- `prometeo.curp.models`, [23](#)
- `prometeo.dian`, [24](#)
- `prometeo.dian.models`, [27](#)
- `prometeo.sat`, [32](#)
- `prometeo.sat.models`, [35](#)

A

ACCEPTED (*prometeo.sat.client.Status* attribute), 34
 Account (*class in prometeo.banking.client*), 15
 Account (*class in prometeo.banking.models*), 19
 Accountant (*class in prometeo.dian.models*), 27
 accountant() (*prometeo.dian.models.CompanyInfo* property), 27
 AcknowledgementResult (*class in prometeo.sat.client*), 33
 AcknowledgementResult (*class in prometeo.sat.models*), 35
 address() (*prometeo.dian.models.Location* property), 28
 address() (*prometeo.dian.models.Numeration* property), 29
 AF (*prometeo.sat.client.Motive* attribute), 34
 AGUASCALIENTES (*prometeo.curp.client.State* attribute), 22
 ALL (*prometeo.sat.client.DocumentType* attribute), 34
 ALL (*prometeo.sat.client.Motive* attribute), 34
 ALL (*prometeo.sat.client.SendType* attribute), 35
 ALL (*prometeo.sat.client.Status* attribute), 34
 amount() (*prometeo.dian.models.Balance* property), 27
 anio_reg() (*prometeo.curp.models.DocumentData* property), 23
 ANY (*prometeo.sat.client.BillStatus* attribute), 34
 approved() (*prometeo.banking.models.PreprocessTransfer* property), 20
 APRIL (*prometeo.dian.client.MonthlyPeriod* attribute), 26
 AUGUST (*prometeo.dian.client.MonthlyPeriod* attribute), 26
 auth_fields() (*prometeo.banking.models.ProviderDetail* property), 20
 Authorization (*prometeo.dian.client.NumerationType* attribute), 25
 authorization_devices() (*prometeo.banking.models.PreprocessTransfer*

property), 20

B

B (*prometeo.sat.client.DocumentType* attribute), 34
 BAJA_CALIFORNIA (*prometeo.curp.client.State* attribute), 22
 BAJA_CALIFORNIA_SUR (*prometeo.curp.client.State* attribute), 22
 Balance (*class in prometeo.dian.models*), 27
 balance() (*prometeo.banking.models.Account* property), 19
 balance_dollar() (*prometeo.banking.models.CreditCard* property), 19
 balance_local() (*prometeo.banking.models.CreditCard* property), 19
 BankingAPIClient (*class in prometeo.banking.client*), 15
 BaseClient (*class in prometeo.base_client*), 21
 BaseSession (*class in prometeo.base_session*), 21
 BillStatus (*class in prometeo.sat.client*), 34
 BIMONTHLY (*prometeo.dian.client.Periodicity* attribute), 25
 BimonthlyPeriod (*class in prometeo.dian.client*), 26
 branch() (*prometeo.banking.models.Account* property), 19

C

C (*prometeo.sat.client.SendType* attribute), 35
 call_api() (*prometeo.base_client.BaseClient* method), 21
 CAMPECHE (*prometeo.curp.client.State* attribute), 22
 CANCELLED (*prometeo.sat.client.BillStatus* attribute), 34
 capital_composition() (*prometeo.dian.models.CompanyInfo* property), 27
 CapitalComposition (*class in prometeo.dian.models*), 27
 CARNE_DIPLOMATICO (*prometeo.dian.client.DocumentType* attribute),

26
 CEDULA_CIUDADANIA (prometeo.dian.client.DocumentType attribute), 26
 CEDULA_EXTRANJERIA (prometeo.dian.client.DocumentType attribute), 26
 CERTIFICADO_REGISTRADURIA (prometeo.dian.client.DocumentType attribute), 26
 certification_date() (prometeo.sat.models.CFDIBill property), 35
 certification_pac() (prometeo.sat.models.CFDIBill property), 35
 CFDI (prometeo.sat.client.LoginScope attribute), 34
 CFDIBill (class in prometeo.sat.models), 35
 CFIDownloadItem (class in prometeo.sat.models), 36
 CHIAPAS (prometeo.curp.client.State attribute), 22
 CHIHUAHUA (prometeo.curp.client.State attribute), 22
 city() (prometeo.dian.models.Location property), 28
 CIUDAD_DE_MEXICO (prometeo.curp.client.State attribute), 22
 clave_entidad_registro() (prometeo.curp.models.DocumentData property), 23
 clave_municipio_registro() (prometeo.curp.models.DocumentData property), 23
 Client (class in prometeo.banking.models), 19
 close_date() (prometeo.banking.models.CreditCard property), 19
 CO (prometeo.sat.client.Motive attribute), 34
 COAHUILA (prometeo.curp.client.State attribute), 22
 code() (prometeo.banking.models.Provider property), 20
 COLIMA (prometeo.curp.client.State attribute), 22
 CompanyInfo (class in prometeo.dian.models), 27
 confirm_transfer() (prometeo.banking.client.BankingAPIClient method), 15
 confirm_transfer() (prometeo.banking.client.Session method), 17
 ConfirmTransfer (class in prometeo.banking.models), 19
 constitution_date() (prometeo.dian.models.CompanyInfo property), 28
 correction_code() (prometeo.dian.models.RentDeclaration property), 30
 correction_code() (prometeo.dian.models.VATDeclaration property), 31
 count() (prometeo.sat.models.CFDIDownloadItem property), 36
 country() (prometeo.banking.models.Provider property), 20
 country() (prometeo.banking.models.ProviderDetail property), 20
 country() (prometeo.dian.models.Location property), 28
 country() (prometeo.dian.models.Numeration property), 29
 credit() (prometeo.banking.models.Movement property), 20
 CreditCard (class in prometeo.banking.client), 17
 CreditCard (class in prometeo.banking.models), 19
 CT (prometeo.sat.client.DocumentType attribute), 34
 CurpAPIClient (class in prometeo.curp.client), 22
 currency() (prometeo.banking.models.Account property), 19

D

date() (prometeo.banking.models.Movement property), 20
 DE (prometeo.sat.client.Motive attribute), 34
 debit() (prometeo.banking.models.Movement property), 20
 DECEMBER (prometeo.dian.client.MonthlyPeriod attribute), 26
 department() (prometeo.dian.models.Location property), 28
 department() (prometeo.dian.models.Numeration property), 29
 detail() (prometeo.banking.models.Movement property), 20
 DianAPIClient (class in prometeo.dian.client), 24
 direction_code() (prometeo.dian.models.RentDeclaration property), 30
 direction_code() (prometeo.dian.models.Retentions property), 31
 direction_code() (prometeo.dian.models.VATDeclaration property), 31
 document() (prometeo.dian.models.Accountant property), 27
 document() (prometeo.dian.models.Member property), 29
 document() (prometeo.dian.models.Representative property), 30
 document_data() (prometeo.curp.models.QueryResult property), 24
 document_type() (prometeo.dian.models.Member property), 29

document_type() (prometeo.dian.models.Representative property), 30

document_type() (prometeo.sat.models.AcknowledgementResult property), 35

DocumentData (class in prometeo.curp.models), 23

DOCUMENTO_IDENTIFICACION_EXTRANJERO (prometeo.dian.client.DocumentType attribute), 26

DOCUMENTO_IDENTIFICACION_EXTRANJERO_PERSONA_JURIDICA (prometeo.dian.client.DocumentType attribute), 26

DocumentType (class in prometeo.dian.client), 26

DocumentType (class in prometeo.sat.client), 34

Download (class in prometeo.base_client), 21

download_emitted_bills() (prometeo.sat.client.Session method), 32

download_received_bills() (prometeo.sat.client.Session method), 32

download_url() (prometeo.sat.models.DownloadFile property), 36

DownloadFile (class in prometeo.sat.models), 36

DownloadRequest (class in prometeo.sat.client), 34

DownloadRequest (class in prometeo.sat.models), 36

due_date() (prometeo.banking.models.CreditCard property), 19

DURANGO (prometeo.curp.client.State attribute), 22

dv() (prometeo.dian.models.Numeration property), 29

dv() (prometeo.dian.models.RentDeclaration property), 30

dv() (prometeo.dian.models.VATDeclaration property), 31

E

economic_activity() (prometeo.dian.models.RentDeclaration property), 30

effect() (prometeo.sat.models.CFDIBill property), 35

email() (prometeo.dian.models.Location property), 28

emitted_date() (prometeo.sat.models.CFDIBill property), 35

emitter_reason() (prometeo.sat.models.CFDIBill property), 35

emitter_rfc() (prometeo.sat.models.CFDIBill property), 35

entidad_registro() (prometeo.curp.models.DocumentData property), 23

ESTADO_DE_MEXICO (prometeo.curp.client.State attribute), 22

extra_data() (prometeo.banking.models.Movement property), 20

F

FC (prometeo.sat.client.Motive attribute), 34

FEBRUARY (prometeo.dian.client.MonthlyPeriod attribute), 26

FEMALE (prometeo.curp.client.Gender attribute), 22

Field (class in prometeo.dian.models), 28

fields() (prometeo.dian.models.RentDeclaration property), 30

fields() (prometeo.dian.models.Retentions property), 31

fields() (prometeo.dian.models.VATDeclaration property), 31

file_name() (prometeo.sat.models.AcknowledgementResult property), 35

finish_login() (prometeo.banking.client.Session method), 17

first_name() (prometeo.dian.models.Name property), 29

first_surname() (prometeo.dian.models.Name property), 29

foja() (prometeo.curp.models.DocumentData property), 23

foreign() (prometeo.dian.models.CapitalComposition property), 27

foreign_private() (prometeo.dian.models.CapitalComposition property), 27

foreign_public() (prometeo.dian.models.CapitalComposition property), 27

form_number() (prometeo.dian.models.RentDeclaration property), 30

form_number() (prometeo.dian.models.Retentions property), 31

form_number() (prometeo.dian.models.VATDeclaration property), 31

G

Gender (class in prometeo.curp.client), 22

get_accounts() (prometeo.banking.client.Session method), 17

get_acknowledgements() (prometeo.sat.client.Session method), 33

get_balances() (prometeo.dian.client.Session method), 24

get_clients() (prometeo.banking.client.Session method), 17

get_company_info() (prometeo.dian.client.Session method), 24

get_credit_cards() (prometeo.banking.client.Session method), 18

`get_download()` (*prometeo.sat.client.AcknowledgementResult* method), 33
`get_download()` (*prometeo.sat.client.DownloadRequest* method), 34
`get_downloads()` (*prometeo.sat.client.Session* method), 33
`get_emitted_bills()` (*prometeo.sat.client.Session* method), 33
`get_file()` (*prometeo.base_client.Download* method), 21
`get_interactive_context()` (*prometeo.banking.client.Session* method), 18
`get_movements()` (*prometeo.banking.client.Account* method), 15
`get_movements()` (*prometeo.banking.client.CreditCard* method), 17
`get_numeration()` (*prometeo.dian.client.Session* method), 25
`get_provider_detail()` (*prometeo.banking.client.BankingAPIClient* method), 16
`get_providers()` (*prometeo.banking.client.BankingAPIClient* method), 16
`get_received_bills()` (*prometeo.sat.client.Session* method), 33
`get_rent_declaration()` (*prometeo.dian.client.Session* method), 25
`get_retentions()` (*prometeo.dian.client.Session* method), 25
`get_session()` (*prometeo.base_client.BaseClient* method), 21
`get_session_key()` (*prometeo.base_session.BaseSession* method), 21
`get_status()` (*prometeo.base_session.BaseSession* method), 21
`get_vat_declaration()` (*prometeo.dian.client.Session* method), 25
GUANAJUATO (*prometeo.curp.client.State* attribute), 22
GUERRERO (*prometeo.curp.client.State* attribute), 23

H

Habilitation (*prometeo.dian.client.NumerationType* attribute), 25
HIDALGO (*prometeo.curp.client.State* attribute), 23

I

`id()` (*prometeo.banking.models.Account* property), 19
`id()` (*prometeo.banking.models.Client* property), 19
`id()` (*prometeo.banking.models.CreditCard* property), 19
`id()` (*prometeo.banking.models.Movement* property), 20
`id()` (*prometeo.banking.models.TransferInstitution* property), 21
`id()` (*prometeo.sat.models.AcknowledgementResult* property), 35
`id()` (*prometeo.sat.models.CFDIBill* property), 36
Inhabilitation (*prometeo.dian.client.NumerationType* attribute), 25
`is_ready()` (*prometeo.sat.client.DownloadRequest* method), 34

J

JALISCO (*prometeo.curp.client.State* attribute), 23
JANUARY (*prometeo.dian.client.MonthlyPeriod* attribute), 26
JANUARY_APRIL (*prometeo.dian.client.QuarterlyPeriod* attribute), 26
JANUARY_FEBRUARY (*prometeo.dian.client.BimonthlyPeriod* attribute), 26
JULY (*prometeo.dian.client.MonthlyPeriod* attribute), 26
JULY_AUGUST (*prometeo.dian.client.BimonthlyPeriod* attribute), 26
JUNE (*prometeo.dian.client.MonthlyPeriod* attribute), 26

L

`libro()` (*prometeo.curp.models.DocumentData* property), 23
`list_transfer_institutions()` (*prometeo.banking.client.BankingAPIClient* method), 16
`list_transfer_institutions()` (*prometeo.banking.client.Session* method), 18
Location (class in *prometeo.dian.models*), 28
`location()` (*prometeo.dian.models.CompanyInfo* property), 28
`login()` (*prometeo.banking.client.BankingAPIClient* method), 16
`login()` (*prometeo.dian.client.DianAPIClient* method), 24
`login()` (*prometeo.sat.client.SatAPIClient* method), 32
LoginScope (class in *prometeo.sat.client*), 34
`logout()` (*prometeo.banking.client.BankingAPIClient* method), 16
`logout()` (*prometeo.banking.client.Session* method), 18
`logout()` (*prometeo.sat.client.Session* method), 33

M

MALE (*prometeo.curp.client.Gender* attribute), 22
MARCH (*prometeo.dian.client.MonthlyPeriod* attribute), 26

- MARCH_APRIL (*prometeo.dian.client.BimonthlyPeriod attribute*), 26
- MAY (*prometeo.dian.client.MonthlyPeriod attribute*), 26
- MAY_AUGUST (*prometeo.dian.client.QuarterlyPeriod attribute*), 26
- MAY_JUNE (*prometeo.dian.client.BimonthlyPeriod attribute*), 26
- Member (*class in prometeo.dian.models*), 28
- members() (*prometeo.dian.models.CompanyInfo property*), 28
- message() (*prometeo.banking.models.ConfirmTransfer property*), 19
- message() (*prometeo.banking.models.PreprocessTransfer property*), 20
- MICHOACAN (*prometeo.curp.client.State attribute*), 23
- MONTHLY (*prometeo.sat.client.Motive attribute*), 34
- MonthlyPeriod (*class in prometeo.dian.client*), 26
- MORELOS (*prometeo.curp.client.State attribute*), 23
- Motive (*class in prometeo.sat.client*), 34
- motive() (*prometeo.sat.models.AcknowledgementResult property*), 35
- Movement (*class in prometeo.banking.models*), 19
- municipality() (*prometeo.dian.models.Numeration property*), 29
- municipio_registro() (*prometeo.curp.models.DocumentData property*), 24
- ## N
- N (*prometeo.sat.client.SendType attribute*), 35
- NACIDO_EN_EL_EXTRANJERO (*prometeo.curp.client.State attribute*), 23
- Name (*class in prometeo.dian.models*), 29
- name() (*prometeo.banking.models.Account property*), 19
- name() (*prometeo.banking.models.Client property*), 19
- name() (*prometeo.banking.models.CreditCard property*), 19
- name() (*prometeo.banking.models.Provider property*), 20
- name() (*prometeo.banking.models.ProviderDetail property*), 20
- name() (*prometeo.banking.models.TransferInstitution property*), 21
- name() (*prometeo.dian.models.Accountant property*), 27
- name() (*prometeo.dian.models.CompanyInfo property*), 28
- name() (*prometeo.dian.models.Field property*), 28
- name() (*prometeo.dian.models.Member property*), 29
- name() (*prometeo.dian.models.Numeration property*), 29
- name() (*prometeo.dian.models.RentDeclaration property*), 30
- name() (*prometeo.dian.models.Representative property*), 30
- name() (*prometeo.dian.models.VATDeclaration property*), 31
- national() (*prometeo.dian.models.CapitalComposition property*), 27
- national_private() (*prometeo.dian.models.CapitalComposition property*), 27
- national_public() (*prometeo.dian.models.CapitalComposition property*), 27
- nationality() (*prometeo.dian.models.Member property*), 29
- NAYARIT (*prometeo.curp.client.State attribute*), 23
- nit() (*prometeo.dian.models.Numeration property*), 29
- nit() (*prometeo.dian.models.RentDeclaration property*), 30
- nit() (*prometeo.dian.models.Retentions property*), 31
- nit() (*prometeo.dian.models.VATDeclaration property*), 31
- NOVEMBER (*prometeo.dian.client.MonthlyPeriod attribute*), 26
- NOVEMBER_DECEMBER (*prometeo.dian.client.BimonthlyPeriod attribute*), 26
- NUEVO_LEON (*prometeo.curp.client.State attribute*), 23
- num_acta() (*prometeo.curp.models.DocumentData property*), 24
- number() (*prometeo.banking.models.Account property*), 19
- number() (*prometeo.banking.models.CreditCard property*), 19
- number() (*prometeo.dian.models.Field property*), 28
- Numeration (*class in prometeo.dian.models*), 29
- NumerationRange (*in module prometeo.dian.models*), 30
- NumerationType (*class in prometeo.dian.client*), 25
- ## O
- OAXACA (*prometeo.curp.client.State attribute*), 23
- OCTOBER (*prometeo.dian.client.MonthlyPeriod attribute*), 26
- on_response() (*prometeo.banking.client.BankingAPIClient method*), 16
- on_response() (*prometeo.base_client.BaseClient method*), 21
- other_names() (*prometeo.dian.models.Name property*), 29
- ## P
- PASAPORTE (*prometeo.dian.client.DocumentType attribute*), 25

- tribute*), 26
 - `pdf()` (*prometeo.curp.models.QueryResult* property), 24
 - `pdf()` (*prometeo.dian.models.CompanyInfo* property), 28
 - `pdf()` (*prometeo.dian.models.RentDeclaration* property), 30
 - `pdf()` (*prometeo.dian.models.Retentions* property), 31
 - `pdf()` (*prometeo.dian.models.VATDeclaration* property), 31
 - `pdf_available()` (*prometeo.dian.models.Numeration* property), 29
 - `pdf_url()` (*prometeo.curp.models.QueryResult* property), 24
 - `pdf_url()` (*prometeo.dian.models.CompanyInfo* property), 28
 - `pdf_url()` (*prometeo.dian.models.Numeration* property), 29
 - `pdf_url()` (*prometeo.dian.models.RentDeclaration* property), 30
 - `pdf_url()` (*prometeo.dian.models.Retentions* property), 31
 - `pdf_url()` (*prometeo.dian.models.VATDeclaration* property), 32
 - `pdf_url()` (*prometeo.sat.models.PdfFile* property), 36
 - `PdfFile` (class in *prometeo.sat.models*), 36
 - `period()` (*prometeo.dian.models.Retentions* property), 31
 - `period()` (*prometeo.dian.models.VATDeclaration* property), 32
 - `period()` (*prometeo.sat.models.AcknowledgementResult* property), 35
 - `Periodicity` (class in *prometeo.dian.client*), 25
 - `personal_data()` (*prometeo.curp.models.QueryResult* property), 24
 - `PersonalData` (in module *prometeo.curp.models*), 24
 - `phone1()` (*prometeo.dian.models.Location* property), 28
 - `phone2()` (*prometeo.dian.models.Location* property), 28
 - `PL` (*prometeo.sat.client.DocumentType* attribute), 34
 - `preprocess_transfer()` (*prometeo.banking.client.BankingAPIClient* method), 16
 - `preprocess_transfer()` (*prometeo.banking.client.Session* method), 18
 - `PreprocessTransfer` (class in *prometeo.banking.models*), 20
 - `previous_form()` (*prometeo.dian.models.RentDeclaration* property), 30
 - `previous_form()` (*prometeo.dian.models.VATDeclaration* property), 32
 - `professional_card()` (*prometeo.dian.models.Accountant* property), 27
 - prometeo.banking* (module), 15
 - prometeo.banking.client* (module), 15
 - prometeo.banking.models* (module), 19
 - prometeo.base_client* (module), 21
 - prometeo.base_session* (module), 21
 - prometeo.curp* (module), 22
 - prometeo.curp.models* (module), 23
 - prometeo.dian* (module), 24
 - prometeo.dian.models* (module), 27
 - prometeo.sat* (module), 32
 - prometeo.sat.models* (module), 35
 - `Provider` (class in *prometeo.banking.models*), 20
 - `ProviderDetail` (class in *prometeo.banking.models*), 20
 - `PUEBLA` (*prometeo.curp.client.State* attribute), 23
- ## Q
- `QUARTERLY` (*prometeo.dian.client.Periodicity* attribute), 25
 - `QuarterlyPeriod` (class in *prometeo.dian.client*), 25
 - `QUERETARO` (*prometeo.curp.client.State* attribute), 23
 - `query()` (*prometeo.curp.client.CurpAPIClient* method), 22
 - `QueryResult` (class in *prometeo.curp.models*), 24
 - `QUINTANA_ROO` (*prometeo.curp.client.State* attribute), 23
- ## R
- `ranges()` (*prometeo.dian.models.Numeration* property), 29
 - `reason()` (*prometeo.dian.models.CompanyInfo* property), 28
 - `reason()` (*prometeo.dian.models.Numeration* property), 30
 - `reason()` (*prometeo.dian.models.RentDeclaration* property), 30
 - `reason()` (*prometeo.dian.models.Retentions* property), 31
 - `reason()` (*prometeo.dian.models.VATDeclaration* property), 32
 - `RECEIVED` (*prometeo.sat.client.Status* attribute), 35
 - `receiver_reason()` (*prometeo.sat.models.CFDIBill* property), 36
 - `receiver_rfc()` (*prometeo.sat.models.CFDIBill* property), 36
 - `reception_date()` (*prometeo.sat.models.AcknowledgementResult* property), 35
 - `reference()` (*prometeo.banking.models.Movement* property), 20
 - `REJECTED` (*prometeo.sat.client.Status* attribute), 35

- RentDeclaration (class in prometeo.dian.models), 30
- representation() (prometeo.dian.models.CompanyInfo property), 28
- representation_type() (prometeo.dian.models.Representative property), 31
- Representative (class in prometeo.dian.models), 30
- request_id() (prometeo.banking.models.PreprocessTransfer property), 20
- request_id() (prometeo.sat.models.CFDIDownloadItem property), 36
- request_id() (prometeo.sat.models.DownloadRequest property), 36
- Retentions (class in prometeo.dian.models), 31
- reverse_query() (prometeo.curp.client.CurpAPIClient method), 22
- ## S
- SAN_LUIS_POTOSI (prometeo.curp.client.State attribute), 23
- SatAPIClient (class in prometeo.sat.client), 32
- second_surname() (prometeo.dian.models.Name property), 29
- select_client() (prometeo.banking.client.Session method), 18
- send_type() (prometeo.sat.models.AcknowledgementResult property), 35
- SendType (class in prometeo.sat.client), 35
- SEPTEMBER (prometeo.dian.client.MonthlyPeriod attribute), 26
- SEPTEMBER_DECEMBER (prometeo.dian.client.QuarterlyPeriod attribute), 26
- SEPTEMBER_OCTOBER (prometeo.dian.client.BimonthlyPeriod attribute), 26
- Session (class in prometeo.banking.client), 17
- Session (class in prometeo.dian.client), 24
- Session (class in prometeo.sat.client), 32
- session_class (prometeo.banking.client.BankingAPIClient attribute), 17
- session_class (prometeo.dian.client.DianAPIClient attribute), 24
- session_class (prometeo.sat.client.SatAPIClient attribute), 32
- SIAT (prometeo.sat.client.LoginScope attribute), 34
- SIN_IDENTIFICACION_EXTRANJERO (prometeo.dian.client.DocumentType attribute), 26
- SINALOA (prometeo.curp.client.State attribute), 23
- SONORA (prometeo.curp.client.State attribute), 23
- start_date() (prometeo.dian.models.Accountant property), 27
- start_date() (prometeo.dian.models.Member property), 29
- start_date() (prometeo.dian.models.Representative property), 31
- State (class in prometeo.curp.client), 22
- Status (class in prometeo.sat.client), 34
- status() (prometeo.sat.models.AcknowledgementResult property), 35
- status() (prometeo.sat.models.CFDIBill property), 36
- success() (prometeo.banking.models.ConfirmTransfer property), 19
- ## T
- TABASCO (prometeo.curp.client.State attribute), 23
- TAMAULIPAS (prometeo.curp.client.State attribute), 23
- TARJETA_EXTRANJERIA (prometeo.dian.client.DocumentType attribute), 26
- TARJETA_IDENTIDAD (prometeo.dian.client.DocumentType attribute), 27
- TLAXCALA (prometeo.curp.client.State attribute), 23
- tomo() (prometeo.curp.models.DocumentData property), 24
- total_value() (prometeo.sat.models.CFDIBill property), 36
- TransferInstitution (class in prometeo.banking.models), 20
- type() (prometeo.dian.models.Balance property), 27
- type() (prometeo.sat.models.CFDIDownloadItem property), 36
- ## V
- VALID (prometeo.sat.client.BillStatus attribute), 34
- value() (prometeo.dian.models.Balance property), 27
- value() (prometeo.dian.models.Field property), 28
- VATDeclaration (class in prometeo.dian.models), 31
- VERACRUZ (prometeo.curp.client.State attribute), 23
- ## X
- XC (prometeo.sat.client.DocumentType attribute), 34
- XF (prometeo.sat.client.DocumentType attribute), 34
- ## Y
- year() (prometeo.dian.models.RentDeclaration property), 30
- year() (prometeo.dian.models.Retentions property), 31

`year()` (*prometeo.dian.models.VATDeclaration* property), [32](#)

`YUCATAN` (*prometeo.curp.client.State* attribute), [23](#)

Z

`ZACATECA` (*prometeo.curp.client.State* attribute), [23](#)