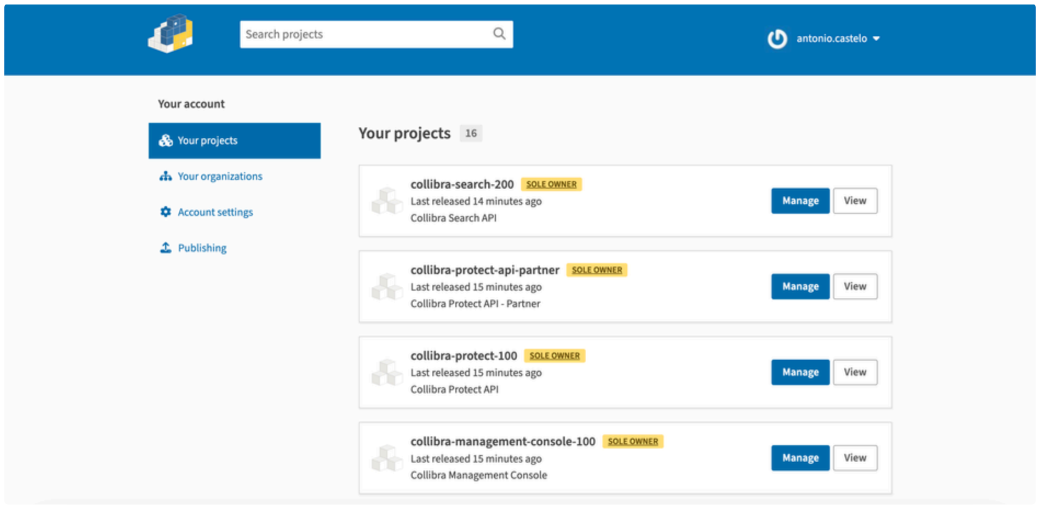
Collibra on Python Package Index

We’ve recently joined the Python Package Index repository with over 500,000 projects and 800,000 users



Your projects 16

collibra-search-200 SOLE OWNER
Last released 14 minutes ago
Collibra Search API

collibra-protect-api-partner SOLE OWNER
Last released 15 minutes ago
Collibra Protect API - Partner

collibra-protect-100 SOLE OWNER
Last released 15 minutes ago
Collibra Protect API

collibra-management-console-100 SOLE OWNER
Last released 15 minutes ago
Collibra Management Console

The Python Package Index (PyPI) is a repository of software for the Python programming language. PyPI helps you find and install software developed and shared by the Python community. Package authors use PyPI to distribute their software. Find, install, publish Python packages with the Python Package Index

```
pip install collibra-core-200
```

Generating your Python client

We’ll be using Swagger codegen.

Swagger Codegen can simplify your build process by generating server stubs and **client SDKs for any API defined with the OpenAPI specification** (formerly known as Swagger), so teams can focus on the API’s implementation and adoption. Please visit the installation section of the Swagger Codegen to learn about how to get the Codegen on your machine.

Take for example the Collibra Core API. The Collibra Core API is the main entry point to interact with your Collibra Data Intelligence Cloud environment. It allows you to create, update or delete all resources such as users, assets, domains or trigger workflows.

Start by downloading the Collibra Core API OpenAPI specification document. Name it `core.json` for ex:

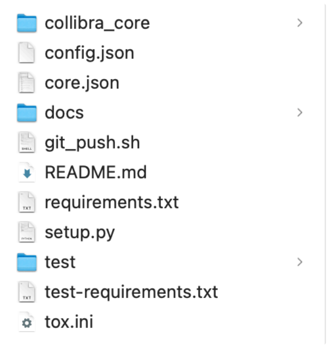
Create a simple `config.json` file containing the project name, package name and version. For example:

```
{
  "projectName":"collibra-core_2.0.0",
  "packageName":"collibra_core",
  "packageVersion":"2.0.0"
}
```

Generate your python client sdk

```
swagger-codegen generate -i core.json -l python -c config.json
```

In the above code, we pass three arguments : `-i` , `-l` and `-c`. `-i` is used to specify the path to your `core.json` API's specification. `-l` is used to specify the language you want to generate. `-c` is used to specify the path to your `config.json` configuration file. Codegen will create the following files for you:



collibra_core
config.json
core.json
docs
git_push.sh
README.md
requirements.txt
setup.py
test
test-requirements.txt
tox.ini

`collibra_core` contains all the generated python client code. `docs` contains all your Collibra Core OpenAPI documentation. `tests` contains your Collibra Core OpenAPI python test code. `requirements.txt` lists all your python client dependencies. `test_requirements.txt` list your tests python dependencies. `README.md` documents your Collibra Core OpenAPI endpoints: installation and usage and getting started. `setup.py` used to build and distribute your package. It typically contains information about your package and looks somewhat like

```python
# coding: utf-8

"""
    Collibra Data Governance Center Core API

    <p>The Core REST API allows you to create your own integrations with Collibra Data Governance Center.</p><p>

    OpenAPI spec version: 2.0

    Generated by: https://github.com/swagger-api/swagger-codegen.git
"""

from setuptools import setup, find_packages  # noqa: H301

NAME = "collibra-core_200"
VERSION = "2.0.0"
# To install the library, run the following
#
# python setup.py install
#
# prerequisite: setuptools
# http://pypi.python.org/pypi/setuptools

REQUIRES = ["urllib3 >= 1.15", "six >= 1.10", "certifi", "python-dateutil"]

setup(
    name=NAME,
    version=VERSION,
    description="Collibra Data Governance Center Core API",
    author_email="",
    url="",
    keywords=["Swagger", "Collibra Data Governance Center Core API"],
    install_requires=REQUIRES,
    packages=find_packages(),
    include_package_data=True,
    long_description="""\
    &lt;p&gt;The Core REST API allows you to create your own integrations with Collibra Data Governance Center.&
    """
)

```

Configuring your Python project

We would suggest changing your `setup.py` file and get the long description from your `README.md` file.

```
# coding: utf-8

"""
    Collibra Data Governance Center Core API

    <p>The Core REST API allows you to create your own integrations with Collibra Data Governance Center.</p><p>

    OpenAPI spec version: 2.0

    Generated by: https://github.com/swagger-api/swagger-codegen.git
"""

from setuptools import setup, find_packages  # noqa: H301

NAME = "collibra-core"
VERSION = "2.0.0"
# To install the library, run the following
#
# python setup.py install
#
# prerequisite: setuptools
# http://pypi.python.org/pypi/setuptools

REQUIRES = ["urllib3 >= 1.15", "six >= 1.10", "certifi", "python-dateutil"]

with open("README.md", "r") as f:
    DESCRIPTION = f.read()

setup(
    name=NAME,
    version=VERSION,
    description="Collibra Data Governance Center Core API",
    keywords=["Swagger", "Collibra Data Governance Center Core API"],
    install_requires=REQUIRES,
    packages=find_packages(),
    include_package_data=True,
    long_description=DESCRIPTION,
    long_description_content_type="text/markdown"
)

```

This document covers some additional details on configuring, packaging and distributing Python projects with `setuptools` that aren't covered in this document. It also assumes that you are already familiar with the contents of the Installing Packages page.

Packaging your Python project

The next step is to generate distribution packages for the package. These are archives that are uploaded to Python Package Index and installed by pip.

```
python setup.py sdist bdist_wheel
```

The `tar.gz` file is a source distribution whereas the `.whl` file is a built distribution. Newer pip versions preferentially install built distributions, but will fall back to source distributions if needed. In this case, our example package is compatible with Python on any platform so only one built distribution is needed. The distributions can be found in the `dist` folder.

Uploading your Python project

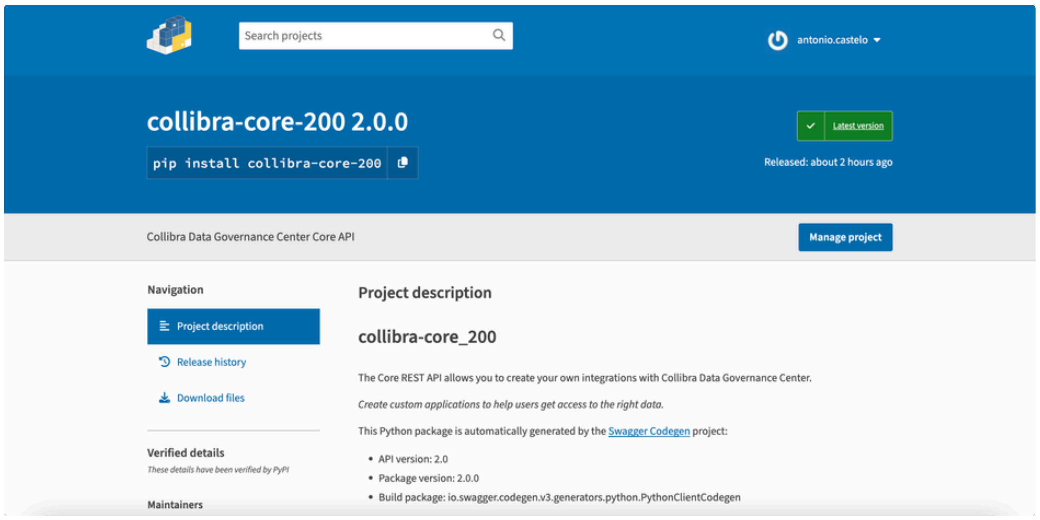
Finally, it's time to upload your package to the Python Package Index. To securely upload your project, you'll need a PyPI API token. Once registered, you can use twine to upload the distribution packages.

```
twine upload dist/*
```

You may wish to upload your package to TestPyPI first and avoid poluting PyPI with wrong packages. The TestPyPI is a separate instance of the PYPI package index intended for testing and experimentation.

```
twine upload --repository testpypi dist/*
```

Once uploaded, your package should be viewable on PyPI and/or TestPyPI; for example



collibra-core-200 2.0.0

```
pip install collibra-core-200
```

Released: about 2 hours ago

Collibra Data Governance Center Core API

Verified details

These details have been verified by PyPI

Maintainers

Project description

collibra-core_200

The Core REST API allows you to create your own integrations with Collibra Data Governance Center.

Create custom applications to help users get access to the right data.

This Python package is automatically generated by the Swagger Codegen project:

- API version: 2.0
- Package version: 2.0.0
- Build package: io.swagger.codegen.v3.generators.python.PythonClientCodegen

Repeat, iterate through all remaining Collibra OpenAPI documents.

collibra-assessments_1.0.0	collibra-catalog_database_registration_1.4.0	collibra-import_2.0.0
collibra-catalog_1.0.0	collibra-catalog_external_profiling_upload_1.0.0	collibra-management_console_1.0.0
collibra-catalog_classification_1.0.0	collibra-catalog_sampling_1.0.0	collibra-protect_1.0.0
collibra-catalog_classification_2.0.0	collibra-catalog_technical_lineage_1.0.0	collibra-protect_api_partner
collibra-catalog_cloud_ingestions_1.0.0	collibra-core_2.0.0	collibra-search_2.0.0

Installing your Python project

You can use pip to install your package and verify that it works. Create a virtual environment and install your package from PyPI or TestPyPI.

```
pip install collibra-core-200
```

```
Collecting collibra-core-200
  Downloading collibra_core_200-2.0.0-py3-none-any.whl.metadata (64 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 64.4/64.4 kB 3.4 MB/s eta 0:00:00
Requirement already satisfied: urllib3>=1.15 in /Users/antonio.castelocollibra.com/opt/anaconda3/lib/python3.9/s
Requirement already satisfied: six>=1.10 in /Users/antonio.castelocollibra.com/opt/anaconda3/lib/python3.9/site-
Requirement already satisfied: certifi in /Users/antonio.castelocollibra.com/opt/anaconda3/lib/python3.9/site-pa
Requirement already satisfied: python-dateutil in /Users/antonio.castelocollibra.com/opt/anaconda3/lib/python3.9
Downloading collibra_core_200-2.0.0-py3-none-any.whl (831 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 831.2/831.2 kB 19.4 MB/s eta 0:00:00
Installing collected packages: collibra-core-200
Successfully installed collibra-core-200-2.0.0

```

Installing from TestPyPI

```
pip install --index-url https://test.pypi.org/simple/  --no-deps collibra-core-200
```

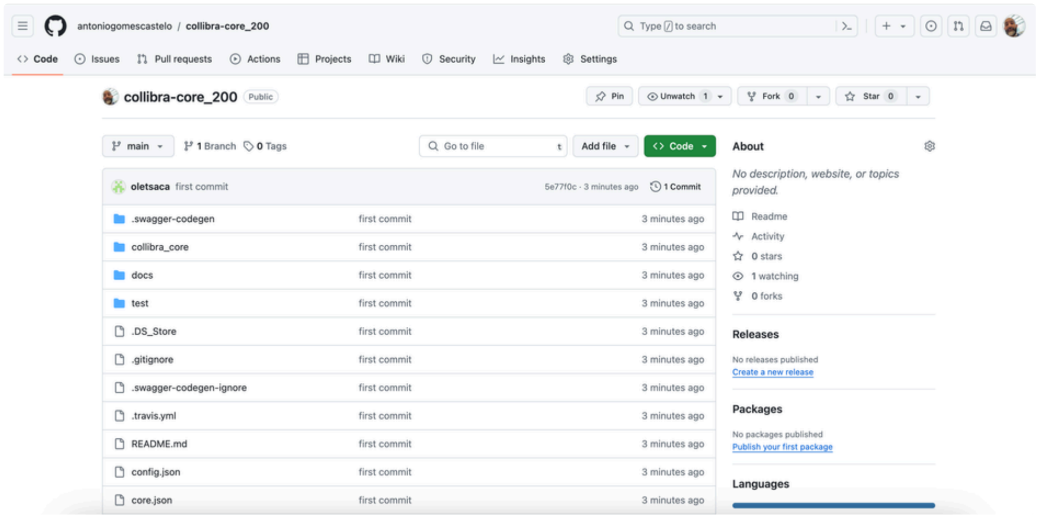
You can test that it was installed correctly by importing the package and running a few API endpoints.

```
import collibra_core

configuration = collibra_core.Configuration()

configuration.host = ENDPOINT

configuration.username = USERNAME

configuration.password = PASSWORD

client = collibra_core.ApiClient(configuration)

assets_api = collibra_core.AssetsApi(client)

assets_api.find_assets()

```

In this document we've shown how to install a package from a PyPI repository but PIP supports installing from various version control systems (VCS).

- Git — git+
- Mercurial — hg+
- Subversion — svn+
- Bazaar — bzr+

Take Git for example



oletsaca first commit 5e77f0c · 3 minutes ago 1 Commit

.swagger-codegen	first commit	3 minutes ago
collibra_core	first commit	3 minutes ago
docs	first commit	3 minutes ago
test	first commit	3 minutes ago
.DS_Store	first commit	3 minutes ago
.gitignore	first commit	3 minutes ago
.swagger-codegen-ignore	first commit	3 minutes ago
.travis.yml	first commit	3 minutes ago
README.md	first commit	3 minutes ago
config.json	first commit	3 minutes ago
core.json	first commit	3 minutes ago

About

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

```
$ pip install collibra-core_200@git+https://github.com/antoniogomescastelo/collibra-core_200
```

```
Collecting collibra-core_200@ git+https://github.com/antoniogomescastelo/collibra-core_200
  Cloning https://github.com/antoniogomescastelo/collibra-core_200  to /private/var/folders/30/s15hbpv55dq3tcc8zx
  Running command git clone --filter=blob:none --quiet https://github.com/antoniogomescastelo/collibra-core_200
  Resolved https://github.com/antoniogomescastelo/collibra-core_200  to commit 5e77f0cdc0f1e0b260ec4c02d9ec171a93
  Preparing metadata (setup.py) ... done
Requirement already satisfied: urllib3>=1.15 in /Users/antonio.castelocollibra.com/opt/anaconda3/lib/python3.9/si
Requirement already satisfied: six>=1.10 in /Users/antonio.castelocollibra.com/opt/anaconda3/lib/python3.9/site-p
Requirement already satisfied: certifi in /Users/antonio.castelocollibra.com/opt/anaconda3/lib/python3.9/site-pac
Requirement already satisfied: python-dateutil in /Users/antonio.castelocollibra.com/opt/anaconda3/lib/python3.9/
```

Stats

We’ll soon be able to view statistics for our projects via Libraries.io, or by using PYPI public dataset on Google BigQuery and learn more about downloads of our packages hosted on PyPI.

stats from July to September 2024:

Downloads	Jul 2024	Aug 2024	Sep 2024