




PyOrthanc: A Python Interface for Orthanc DICOM Servers

Gabriel Couture ¹¶, Yannick Lemaréchal ², and Philippe Després ^{1,2}

1 Centre de recherche de l'Institut universitaire de cardiologie et de pneumologie de Québec-Université Laval, 2725 Ch Ste-Foy, G1V 4G5, Québec, Canada **2** Département de physique, de génie physique et d'optique, Université Laval, 1045, avenue de la Médecine, G1V 0A6, Québec, Canada ¶ Corresponding author

DOI: [10.21105/joss.07562](https://doi.org/10.21105/joss.07562)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Stefan Appelhoff](#)  

Reviewers:

- [@sinharsh](#)
- [@jrkerns](#)
- [@devnums](#)

Submitted: 10 September 2024

Published: 20 February 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

PyOrthanc is an open-source Python library that provides a comprehensive interface for interacting with Orthanc ([Jodogne, 2018](#)), a lightweight, versatile, open-source DICOM server for medical imaging in healthcare and research environments.

Statement of need

Digital Imaging and Communications in Medicine (DICOM) ([Committee, 2020](#)) is the standard for managing and transmitting medical images. Orthanc has gained popularity for its lightweight nature and versatility. However, programmatically interacting with Orthanc servers from its REST API can be complex, especially for those unfamiliar with RESTful APIs. PyOrthanc addresses this challenge by providing a client-side, Pythonic interface to Orthanc servers, abstracting away the complexities of HTTP requests and DICOM data handling. This is in contrast to the Orthanc Python plugin, which offers a powerful means to extend Orthanc's functionality directly within the server environment.

Features and Functionalities

PyOrthanc offers a wide range of features that facilitate data manipulation with Orthanc servers:

1. Orthanc's REST API: PyOrthanc implements Python bindings for all Orthanc REST API routes. All the functionalities exposed in Orthanc are therefore easily accessible from a Pythonic interface.
2. Patient, Study, Series, and Instance: The library provides Python classes that ease data management by following DICOM-level concepts.
3. Querying: PyOrthanc provides utility functions to perform complex queries.
4. Anonymization: PyOrthanc exposes Orthanc's anonymization functionalities, which are useful for research and data sharing.
5. Modalities: Users can easily command Orthanc servers to invoke DICOM Message Service Element (DIMSE) services with other connected modalities, such as Computerized Tomography (CT) scan and hospital Picture Archiving and Communication Systems (PACS), which allows querying, storing, and retrieving DICOM data.
6. Jobs: PyOrthanc allows to monitoring of long-running tasks on the Orthanc server.

7. Orthanc SDK: PyOrthanc provides a mock of the Orthanc SDK, which brings type hints and autocomplete (intelliSense) when working with the Python Plugin.

Architecture and Design

The core of PyOrthanc is the Orthanc class, which handles the connection to the Orthanc server and serves as the entry point for all operations. This class manages authentication, request formatting, and response parsing, presenting a clean API to the user. It uses Python's `httpx` (Coles, 2024) library for HTTP communications. The Orthanc class is used by all the utility functions and classes in PyOrthanc. Note that the class is programmatically generated with `simple-openapi-client` (Couture, 2024) for each new version of the Orthanc REST API.

The library is structured around key DICOM concepts, with main classes representing DICOM-level entities such as patients, studies, series, and instances. These classes provide intuitive methods for common operations, such as querying level-specific information, launching anonymization jobs, and downloading data. PyOrthanc also provides utility classes and functions, useful to monitor long-running tasks within Orthanc or perform complex queries.

Use Cases and Applications

PyOrthanc finds applications in various medical imaging scenarios:

1. Research Data Management: Researchers can use PyOrthanc to automate the process of collecting, anonymizing, and organizing DICOM data.
2. Quality Assurance: The library can be used to develop scripts for checking DICOM data and metadata consistency across large datasets.
3. Clinical Workflow Integration: PyOrthanc can be integrated into clinical workflows, automating tasks such as routing studies to appropriate specialists or triggering analysis pipelines.

Performance and Scalability

PyOrthanc is designed to handle large-scale DICOM operations efficiently. The library implements connection pooling and supports asynchronous operations for improved performance when dealing with multiple concurrent requests.

Conclusion

PyOrthanc provides a powerful and user-friendly interface to Orthanc DICOM servers, enabling Python developers to easily integrate DICOM functionality into their applications. By abstracting the complexities of the Orthanc REST API, PyOrthanc accelerates the development of medical imaging applications and workflows. As the field of medical imaging informatics continues to evolve, tools like PyOrthanc play an important role in advancing research and improving clinical practice by making the complex task of data management easier.

References

- Coles, T. (2024). *HTTPX: A next-generation HTTP client for python*. <https://github.com/encode/httpx>.
- Committee, D. S. (2020). *Digital imaging and communications in medicine (DICOM) standard*. NEMA.

- Couture, G. (2024). *Simple-openapi-client: OpenAPI python client generator that follows the KISS principle*. <https://github.com/gacou54/openapi-client>.
- Jodogne, S. (2018). The orthanc ecosystem for medical imaging. *Journal of Digital Imaging*, 31(3), 341–352. <https://doi.org/10.1007/s10278-018-0082-y>