

---

# **DICOMweb Client Documentation**

*Release 0.40.1*

**Markus D. Herrmann**

**Jul 04, 2020**



---

## Contents:

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation guide</b>	<b>3</b>
2.1	Requirements . . . . .	3
2.2	Installation . . . . .	3
<b>3</b>	<b>User guide</b>	<b>5</b>
3.1	Application Programming Interface (API) . . . . .	5
3.2	Command Line Interface (CLI) . . . . .	13
<b>4</b>	<b>Developer guide</b>	<b>15</b>
4.1	Pull requests . . . . .	15
4.2	Coding style . . . . .	15
4.3	Running tests . . . . .	16
4.4	Building documentation . . . . .	16
<b>5</b>	<b>Conformance statement</b>	<b>17</b>
5.1	QIDO-RS . . . . .	17
5.2	WADO-RS . . . . .	17
5.3	STOW-RS . . . . .	17
<b>6</b>	<b>License</b>	<b>19</b>
<b>7</b>	<b>API documentation</b>	<b>21</b>
7.1	dicomweb_client package . . . . .	21
<b>8</b>	<b>Indices and tables</b>	<b>39</b>
	<b>Python Module Index</b>	<b>41</b>
	<b>Index</b>	<b>43</b>



# CHAPTER 1

---

## Introduction

---

The `dicomweb-client` build distribution provides client interfaces for DICOMweb RESTful services QIDO-RS, WADO-RS and STOW-RS to search, retrieve and store DICOM objects over the web, respectively. For more information about DICOMweb please refer to the documentation of the [DICOM standard](#), in particular [PS3.18](#).

The `dicomweb_client` Python package exposes

- Application Programming Interface (API) (see `api` module)
- Command Line Interface (CLI) (see `cli` module)



### 2.1 Requirements

- Python (version 3.5 or higher)
- Python package manager `pip`

For support of image formats:

- JPEG (`libjpeg` or `libjpeg-turbo`)
- JPEG2000 (`openjpeg`)
- PNG (`libpng`)

### 2.2 Installation

Pre-build package available at PyPi:

```
pip install dicomweb-client
```

Source code available at Github:

```
git clone https://github.com/mghcomputationalpathology/dicomweb-client ~/dicomweb-  
→client  
pip install ~/dicomweb-client
```





The client can be used with any DICOMweb server, such as `dcm4che`, `orthanc` or `DICOMcloud`.

## 3.1 Application Programming Interface (API)

To interact with a publicly accessible server, you only need to provide the `url` for the server address.

```
from dicomweb_client.api import DICOMwebClient

client = DICOMwebClient(url="https://mydicomwebserver.com")
```

Some servers expose the different DICOMweb RESTful services using different path prefixes. For example, the publicly accessible `DICOMcloud` server uses the prefixes `"qidors"`, `"wadors"`, and `"stowrs"` for QIDO-RS, WADO-RS, and STOW-RS, respectively. You can specify these prefixes using `qido_url_prefix`, `wado_url_prefix`, and `stow_url_prefix`.

```
from dicomweb_client.api import DICOMwebClient

client = DICOMwebClient(
    url="https://dicomcloud.azurewebsites.net",
    qido_url_prefix="qidors",
    wado_url_prefix="wadors",
    stow_url_prefix="stowrs"
)
```

### 3.1.1 Authentication and authorization

To interact with servers requiring authentication, `DICOMwebClient` accepts arbitrary authentication handlers derived from `requests.auth.AuthBase` (see [here](#) for details).

```
from requests.auth import HTTPBasicAuth
from dicomweb_client.api import DICOMwebClient
from dicomweb_client.session_utils import create_session_from_auth

auth = HTTPBasicAuth('myusername', 'mypassword')
session = create_session_from_auth(auth)

client = DICOMwebClient(
    url="https://mydicomwebserver.com",
    session=session
)
```

To simplify usage for basic HTTP authentication, you may also directly provide a username and password using the corresponding arguments.

```
from dicomweb_client.api import DICOMwebClient
from dicomweb_client.session_utils import create_session_from_user_pass

session = create_session_from_user_pass(
    username='myusername',
    password='mypassword'
)

client = DICOMwebClient(
    url="https://mydicomwebserver.com",
    session=session
)
```

To interact with servers supporting token-based authorization, you can provide the access token using the headers argument (the header will be included in every client request message).

```
from dicomweb_client.api import DICOMwebClient

access_token = "mytoken"
client = DICOMwebClient(
    url="https://mydicomwebserver.com",
    headers={"Authorization": "Bearer {}".format(access_token)}
)
```

To interact with servers requiring certificate-based authentication, you can provide the CA bundle and client certificate using the `ca_bundle` and `cert` arguments, respectively.

```
from dicomweb_client.api import DICOMwebClient
from dicomweb_client.session_utils import (
    create_session,
    add_certs_to_session
)

session = create_session()
session = add_certs_to_session(
    session=session,
    ca_bundle="/path/to/ca.crt",
    cert="/path/to/cert.pem"
)

client = DICOMwebClient(url="https://mydicomwebserver.com")
```

To interact with a server of the Google Healthcare API requiring OpenID Connect based authentication and authoriza-

tion, provide a session authenticated using the Google Cloud Platform (GCP) credentials. See [GCP documentation](#) for details.

Note that GCP authentication requires installation of the package distribution with the `gcp` extra requirements: `$ pip install dicomweb-client[gcp]`.

```
from dicomweb_client.api import DICOMwebClient
from dicomweb_client.session_utils import create_session_from_gcp_credentials

session = create_session_from_gcp_credentials()

client = DICOMwebClient(
    url="https://mydicomwebserver.com",
    session=session
)
```

### 3.1.2 STOW-RS StoreInstances

Store a single dataset obtained from a PS3.10 file:

```
from dicomweb_client.api import DICOMwebClient
import pydicom

filename = "/path/to/file.dcm"
dataset = pydicom.dcmread(filename)
client.store_instances(datasets=[dataset])
```

### 3.1.3 QIDO-RS SearchForStudies

Search for all studies (up to server-defined maximum set per call - see below to iteratively get all studies):

```
studies = client.search_for_studies()
```

Search for studies filtering by *PatientID*:

```
studies = client.search_for_studies(search_filters={'PatientID': 'ABC123'})
```

Note that attributes can be specified in `search_filters` using either the keyword or the tag:

```
studies = client.search_for_studies(search_filters={'00100020': 'ABC123'})
```

Search for all studies but limit the number of returned results using the `limit` parameter.

```
studies_subset = client.search_for_studies(limit=100)
```

A server may also automatically limit the number of results that it returns per search request. In this case, the method can be called repeatedly to request remaining results using the `offset` parameter.

```
studies = []
offset = 0
while True:
    subset = client.search_for_studies(offset=offset)
    if len(subset) == 0:
        break
```

(continues on next page)

(continued from previous page)

```
studies.extend(subset)
offset += len(subset)
```

### 3.1.4 QIDO-RS SearchForSeries

Search for all series:

```
series = client.search_for_series()
```

Search for series of a given study:

```
series = client.search_for_series('1.2.826.0.1.3680043.8.1055.1.20111103111148288.
↳98361414.79379639')
```

Search for series filtering by *AccessionNumber*:

```
series = client.search_for_series(search_filters={'AccessionNumber': '123456'})
```

Search for series filtering by *AccessionNumber* (using wildcard ? to match a range of numbers):

```
series = client.search_for_series(search_filters={'AccessionNumber': '12345?'})
```

Search for series filtering by *SeriesDescription*:

```
series = client.search_for_series(search_filters={'SeriesDescription': 'T2 AXIAL'})
```

Search for series filtering by *SeriesDescription* (using wildcard \* to match a range of descriptions):

```
series = client.search_for_series(search_filters={'SeriesDescription': 'T2 AX*'})
```

Search for series filtering by *Modality*:

```
series = client.search_for_series(search_filters={'Modality': 'CT'})
```

### 3.1.5 QIDO-RS SearchForInstances

Search for all instances:

```
instances = client.search_for_instances()
```

Search for instances of a given study and series:

```
instances = client.search_for_instances(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↳79379639',
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↳24517034'
)
```

Search for instances filtering by *SOPClassUID*:

```
instances = client.search_for_instances(search_filters={'SOPClassUID': '1.2.840.10008.
↳5.1.4.1.1.2'})
```

### 3.1.6 WADO-RS RetrieveStudy

Retrieve instances of a given study:

```
instances = client.retrieve_study('1.2.826.0.1.3680043.8.1055.1.20111103111148288.
↳98361414.79379639')
```

### 3.1.7 WADO-RS RetrieveSeries

Retrieve instances of a given series:

```
instances = client.retrieve_series(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↳79379639',
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↳24517034'
)
```

Retrieve full instances of a given series using specific JPEG 2000 transfer syntax for encoding of bulk data:

```
instance = client.retrieve_instance(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↳79379639',
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↳24517034'
    media_types=(('application/dicom', '1.2.840.10008.1.2.4.90', ), )
)
```

Retrieve bulk data of instances of a given series using specific JPEG 2000 transfer syntax:

```
instance = client.retrieve_instance(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↳79379639',
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↳24517034'
    media_types=(('image/jp2', '1.2.840.10008.1.2.4.90', ), )
)
```

### 3.1.8 WADO-RS RetrieveInstance

Retrieve full instance using default Explicit VR Little Endian transfer syntax for encoding of bulk data:

```
instance = client.retrieve_instance(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↳79379639',
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↳24517034'
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534
↳'
)
```

Retrieve full instance using specific JPEG 2000 transfer syntax for encoding of bulk data:

```
instance = client.retrieve_instance(  
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.  
↪79379639',  
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.  
↪24517034'  
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534  
↪',  
    media_types=(('application/dicom', '1.2.840.10008.1.2.4.90', ), )  
)
```

Retrieve bulk data of instance using specific JPEG 2000 transfer syntax:

```
instance = client.retrieve_instance(  
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.  
↪79379639',  
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.  
↪24517034'  
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534  
↪',  
    media_types=(('image/jp2', '1.2.840.10008.1.2.4.90', ), )  
)
```

### 3.1.9 WADO-RS RetrieveMetadata

Retrieve metadata for instances of a given study:

```
metadata = client.retrieve_study_metadata('1.2.826.0.1.3680043.8.1055.1.  
↪20111103111148288.98361414.79379639')
```

Retrieve metadata for instances of a given series:

```
metadata = client.retrieve_series_metadata(  
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.  
↪79379639',  
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.  
↪24517034'  
)
```

Retrieve metadata for a particular instance:

```
metadata = client.retrieve_instance_metadata(  
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.  
↪79379639',  
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.  
↪24517034'  
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534  
↪'  
)
```

---

**Note:** WADO-RS RetrieveMetadata always returns metadata at the instance-level, retrieve\_study\_metadata() and retrieve\_series\_metadata() return an array of metadata items for each instance belonging to a given study and series, respectively.

---

### 3.1.10 WADO-RS RetrieveFrames

Retrieve a set of frames with default transfer syntax (“application/octet-stream”):

```
frames = client.retrieve_instance_frames(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↪79379639'
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↪24517034'
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534
↪'
    frame_numbers=[1, 2]
)
```

Retrieve a set of frames of a given instances as JPEG compressed image:

```
frames = client.retrieve_instance_frames(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↪79379639'
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↪24517034'
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534
↪'
    frame_numbers=[1, 2],
    media_types=('image/jpeg', )
)
```

Retrieve a set of frames of a given instances as compressed image in any available format:

```
frames = client.retrieve_instance_frames(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↪79379639'
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↪24517034'
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534
↪'
    frame_numbers=[1, 2],
    media_types=('image/*', )
)
```

Retrieve a set of frames of a given instances as either JPEG 2000 or JPEG-LS compressed image:

```
frames = client.retrieve_instance_frames(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↪79379639'
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↪24517034'
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534
↪'
    frame_numbers=[1, 2],
    media_types=('image/jp2', 'image/x-jpls', )
)
```

Retrieve a set of frames of a given instances as either JPEG, JPEG 2000 or JPEG-LS lossless compressed image using specific transfer syntaxes:

```
frames = client.retrieve_instance_frames(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↪79379639'
```

(continues on next page)

(continued from previous page)

```

series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↪24517034'
sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534
↪'
frame_numbers=[1, 2],
media_types=(
    ('image/jpeg', '1.2.840.10008.1.2.4.57', ),
    ('image/jp2', '1.2.840.10008.1.2.4.90', ),
    ('image/x-jpls', '1.2.840.10008.1.2.4.80', ),
)
)

```

### 3.1.11 WADO-RS RetrieveBulkdata

Retrieve bulk data given a URL:

```
data = client.retrieve_bulkdata('https://mydicomwebserver.com/studies/...')
```

### 3.1.12 WADO-RS RetrieveRenderedTransaction

Retrieve a single-frame image instance rendered as a PNG compressed image:

```

frames = client.retrieve_instance_rendered(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↪79379639'
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↪24517034'
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534
↪'
    media_types=('image/png', )
)

```

Retrieve a single frame of a multi-frame image instance rendered as a high-quality JPEG compressed image that includes an ICC profile:

```

frames = client.retrieve_instance_frames_rendered(
    study_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.
↪79379639'
    series_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.
↪24517034'
    sop_instance_uid='1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534
↪'
    frame_numbers=[1],
    media_types=('image/jpeg', ),
    params={'quality': 95, 'iccprofile': 'yes'}
)

```

When frames are retrieved in image format, they can be converted into a *NumPy* array using the *PIL* module:

```

from io import BytesIO

import numpy as np
from PIL import Image

```

(continues on next page)



(continued from previous page)

```
image = Image.open(BytesIO(frames[0]))
array = np.array(image)
```

**Warning:** Retrieving images using lossy compression methods may lead to image recompression artifacts if the images have been stored lossy compressed.

### 3.1.13 Loading JSON Data To pydicom

Load metadata from JSON format into a `pydicom.dataset.Dataset` object. A common use for this is translating metadata received from a `RetrieveMetadata` or a `SearchFor`-style request:

```
from dicomweb_client.api import load_json_dataset

metadata = client.retrieve_study_metadata('1.2.826.0.1.3680043.8.1055.1.
↪20111103111148288.98361414.79379639')
metadata_datasets = [load_json_dataset(ds) for ds in metadata]
```

## 3.2 Command Line Interface (CLI)

Search for studies:

```
dicomweb_client --url https://dicomcloud.azurewebsites.net/qidors search studies
```

Retrieve metadata for all instances of a given study:

```
dicomweb_client --url https://dicomcloud.azurewebsites.net/wadors \
  retrieve studies \
  --study 1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.79379639 \
  metadata
```

The output can be *dicomized* for human interpretation:

```
dicomweb_client --url https://dicomcloud.azurewebsites.net/wadors \
  retrieve studies \
  --study 1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.79379639 \
  metadata \
  --dicomize
```

Retrieve the full Part 3.10 files for all instances of a given study:

```
dicomweb_client --url https://dicomcloud.azurewebsites.net/wadors \
  retrieve studies \
  --study 1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.79379639 \
  full
```

Retrieve a single frame of a given instances as JPEG compressed image:

```
dicomweb_client --url https://dicomcloud.azurewebsites.net/wadors \
  retrieve instances \
```

(continues on next page)

(continued from previous page)

```
--study 1.2.826.0.1.3680043.8.1055.1.20111103111148288.98361414.79379639 \  
--series 1.2.826.0.1.3680043.8.1055.1.20111103111208937.49685336.24517034 \  
--instance 1.2.826.0.1.3680043.8.1055.1.20111103111208937.40440871.13152534 \  
frames \  
--numbers 1 \  
--media-type image/jpeg
```

Store instances to a Google DICOMweb store:

```
dicomweb_client --url https://healthcare.googleapis.com/v1beta1/projects/MYPROJECT/  
→locations/us-central1/datasets/MYDATASET/dicomStores/MYDICOMSTORE/dicomWeb \  
--token $(gcloud auth print-access-token) \  
store instances \  
dicomfiles/*
```

Source code is available at Github and can be cloned via git:

```
git clone https://github.com/mghcomputationalpathology/dicomweb-client ~/dicomweb-  
client
```

The `dicomweb_client` package can be installed in *develop* mode for local development:

```
pip install -e ~/dicomweb-client
```

## 4.1 Pull requests

Don't commit code changes to the `master` branch. New features should be implemented in a separate branch called `feature/*` and bug fixes should be applied in separate branch called `bugfix/*`.

Before creating a pull request on Github, read the coding style guideline, run the tests and check PEP8 compliance.

## 4.2 Coding style

Code must comply with [PEP 8](#). The `flake8` package is used to enforce compliance.

The project uses `numpydoc` for documenting code according to [PEP 257](#) docstring conventions. Further information and examples for the NumPy style can be found at the [NumPy Github repository](#) and the website of the [Napoleon sphinx extension](#).

All API classes, functions and modules must be documented (including “private” functions and methods). Each docstring must describe input parameters and return values. Types must be specified using type hints as specified by [PEP 484](#) (see `typing` module) in both the function definition as well as the docstring.

## 4.3 Running tests

The project uses `pytest` to write and runs unit tests. Tests should be placed in a separate `tests` folder within the package root folder. Files containing actual test code should follow the pattern `test_*.py`.

Install requirements:

```
pip install -r ~/dicomweb-client/requirements_test.txt
```

Run tests (including checks for PEP8 compliance):

```
cd ~/dicomweb-client
pytest --flake8
```

## 4.4 Building documentation

Install requirements:

```
pip install -r ~/dicomweb-client/requirements_docs.txt
```

Build documentation in *HTML* format:

```
cd ~/dicomweb-client
sphinx-build -b html docs/ docs/build/
```

The built `index.html` file will be located in `docs/build`.

---

 Conformance statement
 

---

## 5.1 QIDO-RS

Method	Resource	Implemented
GET	SearchForStudies	Y
GET	SearchForSeries	Y
GET	SearchForInstances	Y

## 5.2 WADO-RS

Method	Resource	Implemented
GET	RetrieveStudy	Y
GET	RetrieveSeries	Y
GET	RetrieveInstance	Y
GET	RetrieveMetadata	Y*
GET	RetrieveBulkdata	Y
GET	RetrieveFrames	Y
GET	RetrieveRenderedTransaction	Y

- *Metadata* resource representations are requested in JSON format according to the [DICOM JSON model](#) using `application/dicom+json` media type. Retrieval of metadata in XML form using `application/dicom+xml` is not supported.

## 5.3 STOW-RS

Method	Resource	Implemented
POST	StoreInstances	Y



## CHAPTER 6

---

### License

---

*DICOMweb Client* is free and open source software licensed under the permissive [MIT license](#).





## 7.1 dicomweb\_client package

### 7.1.1 dicomweb\_client.api module

Application Programming Interface (API).

```
class dicomweb_client.api.DICOMwebClient (url, session=None, qido_url_prefix=None,  
wado_url_prefix=None, stow_url_prefix=None,  
delete_url_prefix=None, proxies=None, headers=None, callback=None, chunk_size=None)
```

Bases: object

Class for connecting to and interacting with a DICOMweb RESTful service.

**base\_url**

unique resource locator of the DICOMweb service

**Type** str

**protocol**

name of the protocol, e.g. "https"

**Type** str

**host**

IP address or DNS name of the machine that hosts the server

**Type** str

**port**

number of the port to which the server listens

**Type** int

**url\_prefix**

URL path prefix for DICOMweb services (part of *base\_url*)

Type str

**qido\_url\_prefix**

URL path prefix for QIDO-RS (not part of *base\_url*)

Type Union[str, None]

**wado\_url\_prefix**

URL path prefix for WADO-RS (not part of *base\_url*)

Type Union[str, None]

**stow\_url\_prefix**

URL path prefix for STOW-RS (not part of *base\_url*)

Type Union[str, None]

**delete\_url\_prefix**

URL path prefix for DELETE (not part of *base\_url*)

Type Union[str, None]

**Parameters**

- **url** (*str*) – base unique resource locator consisting of protocol, hostname (IP address or DNS name) of the machine that hosts the server and optionally port number and path prefix
- **session** (*requests.Session*, *optional*) – session required to make connection to the DICOMweb service (see *session\_utils.py* to create a valid session if necessary)
- **qido\_url\_prefix** (*str*, *optional*) –
- **qido\_url\_prefix** – URL path prefix for QIDO RESTful services
- **wado\_url\_prefix** (*str*, *optional*) – URL path prefix for WADO RESTful services
- **stow\_url\_prefix** (*str*, *optional*) – URL path prefix for STOW RESTful services
- **delete\_url\_prefix** (*str*, *optional*) – URL path prefix for DELETE RESTful services
- **proxies** (*Dict[str, str]*, *optional*) – mapping of protocol or protocol + host to the URL of a proxy server
- **headers** (*Dict[str, str]*, *optional*) – custom headers that should be included in request messages, e.g., authentication tokens
- **callback** (*Callable*, *optional*) – callback function to manipulate responses generated from requests (see [requests event hooks](#))
- **chunk\_size** (*int*, *optional*) – maximum number of bytes per data chunk using chunked transfer encoding (helpful for storing and retrieving large objects or large collections of objects such as studies or series)

**delete\_instance** (*study\_instance\_uid*, *series\_instance\_uid*, *sop\_instance\_uid*)

Deletes specified instance.

**Parameters**

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier
- **sop\_instance\_uid** (*str*) – unique instance identifier

**Returns** HTTP response object returned.

**Return type** requests.models.Response

---

**Note:** The Delete Instance resource is not part of the DICOM standard and may not be supported by all origin servers.

---

**Warning:** This method performs a DELETE and should be used with caution.

**delete\_series** (*study\_instance\_uid*, *series\_instance\_uid*)

Deletes specified series and its respective instances.

**Parameters**

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier

---

**Note:** The Delete Series resource is not part of the DICOM standard and may not be supported by all origin servers.

---

**Returns** HTTP response object returned.

**Return type** requests.models.Response

**Warning:** This method performs a DELETE and should be used with caution.

**delete\_study** (*study\_instance\_uid*)

Deletes specified study and its respective instances.

**Parameters** **study\_instance\_uid** (*str*) – unique study identifier

**Returns** HTTP response object returned.

**Return type** requests.models.Response

---

**Note:** The Delete Study resource is not part of the DICOM standard and may not be supported by all origin servers.

---

**Warning:** This method performs a DELETE and should be used with caution.

**static lookup\_keyword** (*tag*)

Looks up the keyword of a DICOM attribute.

**Parameters** **tag** (*Union[str, int, Tuple[str, str], pydicom.tag.Tag]*) – attribute tag (e.g. "00080018")

**Returns** attribute keyword (e.g. "SOPInstanceUID")

**Return type** str

**static lookup\_tag** (*keyword*)

Looks up the tag of a DICOM attribute.

**Parameters** **keyword** (*str*) – attribute keyword (e.g. "SOPInstanceUID")

**Returns** attribute tag as HEX string (e.g. "00080018")

**Return type** *str*

**retrieve\_bulkdata** (*url, media\_types=None, byte\_range=None*)

Retrieves bulk data from a given location.

**Parameters**

- **url** (*str*) – location of the bulk data
- **media\_types** (*Tuple[Union[*str*, Tuple[*str*, *str*]]], optional*) – acceptable media types and optionally the UIDs of the corresponding transfer syntaxes
- **byte\_range** (*Tuple[int], optional*) – start and end of byte range

**Returns** bulk data items

**Return type** List[bytes]

**retrieve\_instance** (*study\_instance\_uid, series\_instance\_uid, sop\_instance\_uid, media\_types=None*)

Retrieves an individual DICOM instance.

**Parameters**

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier
- **sop\_instance\_uid** (*str*) – unique instance identifier
- **media\_types** (*Tuple[Union[*str*, Tuple[*str*, *str*]]], optional*) – acceptable media types and optionally the UIDs of the corresponding transfer syntaxes

**Returns** data set

**Return type** pydicom.dataset.Dataset

**retrieve\_instance\_frames** (*study\_instance\_uid, series\_instance\_uid, sop\_instance\_uid, frame\_numbers, media\_types=None*)

Retrieves one or more frames of an individual DICOM instance.

**Parameters**

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier
- **sop\_instance\_uid** (*str*) – unique instance identifier
- **frame\_numbers** (*Sequence[int]*) – one-based positional indices of the frames within the instance
- **media\_types** (*Tuple[Union[*str*, Tuple[*str*, *str*]]], optional*) – acceptable media types and optionally the UIDs of the corresponding transfer syntaxes

**Returns** pixel data for each frame

**Return type** List[bytes]

**retrieve\_instance\_frames\_rendered** (*study\_instance\_uid*, *series\_instance\_uid*,  
*sop\_instance\_uid*, *frame\_numbers*, *media\_types=None*, *params=None*)

Retrieves one or more server-side rendered frames of an individual DICOM instance.

#### Parameters

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier
- **sop\_instance\_uid** (*str*) – unique instance identifier
- **frame\_numbers** (*Sequence[int]*) – one-based positional index of the frame within the instance
- **media\_types** (*Tuple[Union[str, Tuple[str, str]]]*, *optional*) – acceptable media type (choices: "image/jpeg", "image/jp2", "image/gif", "image/png")
- **params** (*Dict[str]*, *optional*) – additional parameters relevant for given *media\_type*, e.g., {"quality": 95} for "image/jpeg" media type

**Returns** rendered frames

**Return type** bytes

---

**Note:** Not all media types are compatible with all SOP classes.

---

**retrieve\_instance\_metadata** (*study\_instance\_uid*, *series\_instance\_uid*, *sop\_instance\_uid*)

Retrieves metadata of an individual DICOM instance.

#### Parameters

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier
- **sop\_instance\_uid** (*str*) – unique instance identifier

**Returns** metadata in DICOM JSON format

**Return type** Dict[str, dict]

**retrieve\_instance\_rendered** (*study\_instance\_uid*, *series\_instance\_uid*, *sop\_instance\_uid*, *media\_types=None*, *params=None*)

Retrieves an individual, server-side rendered DICOM instance.

#### Parameters

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier
- **sop\_instance\_uid** (*str*) – unique instance identifier
- **media\_types** (*Tuple[Union[str, Tuple[str, str]]]*, *optional*) – acceptable media types (choices: "image/jpeg", "image/jp2", "image/gif", "image/png", "video/gif", "video/mp4", "video/h265", "text/html", "text/plain", "text/xml", "text/rtf", "application/pdf")
- **params** (*Dict[str]*, *optional*) – additional parameters relevant for given *media\_type*, e.g., {"quality": 95} for "image/jpeg"

**Returns** rendered instance

**Return type** bytes

**retrieve\_series** (*study\_instance\_uid*, *series\_instance\_uid*, *media\_types=None*)

Retrieves instances of a given DICOM series.

**Parameters**

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier
- **media\_types** (*Tuple[Union[str, Tuple[str, str]]], optional*) – acceptable media types and optionally the UIDs of the corresponding transfer syntaxes

**Returns** data sets

**Return type** List[pydicom.dataset.Dataset]

**retrieve\_series\_metadata** (*study\_instance\_uid*, *series\_instance\_uid*)

Retrieves metadata for instances of a given DICOM series.

**Parameters**

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier

**Returns** metadata in DICOM JSON format

**Return type** Dict[str, dict]

**retrieve\_series\_rendered** (*study\_instance\_uid*, *series\_instance\_uid*, *media\_types=None*,  
*params=None*)

Retrieves an individual, server-side rendered DICOM series.

**Parameters**

- **study\_instance\_uid** (*str*) – unique study identifier
- **series\_instance\_uid** (*str*) – unique series identifier
- **media\_types** (*Tuple[Union[str, Tuple[str, str]]], optional*) – acceptable media types (choices: "image/jpeg", "image/jp2", "image/gif", "image/png", "video/gif", "video/mp4", "video/h265", "text/html", "text/plain", "text/xml", "text/rtf", "application/pdf")
- **params** (*Dict[str, Any], optional*) – additional parameters relevant for given *media\_type*, e.g., {"quality": 95} for "image/jpeg"

**Returns** rendered series

**Return type** bytes

**retrieve\_study** (*study\_instance\_uid*, *media\_types=None*)

Retrieves instances of a given DICOM study.

**Parameters**

- **study\_instance\_uid** (*str*) – unique study identifier
- **media\_types** (*Tuple[Union[str, Tuple[str, str]]], optional*) – acceptable media types and optionally the UIDs of the corresponding transfer syntaxes

**Returns** data sets

**Return type** List[pydicom.dataset.Dataset]

**retrieve\_study\_metadata** (*study\_instance\_uid*)

Retrieves metadata of instances of a given DICOM study.

**Parameters** **study\_instance\_uid** (*str*) – unique study identifier

**Returns** metadata in DICOM JSON format

**Return type** List[Dict[str, dict]]

**search\_for\_instances** (*study\_instance\_uid=None, series\_instance\_uid=None, fuzzymatching=None, limit=None, offset=None, fields=None, search\_filters=None*)

Searches for DICOM instances.

**Parameters**

- **study\_instance\_uid** (*str, optional*) – unique study identifier
- **series\_instance\_uid** (*str, optional*) – unique series identifier
- **fuzzymatching** (*bool, optional*) – whether fuzzy semantic matching should be performed
- **limit** (*int, optional*) – maximum number of results that should be returned
- **offset** (*int, optional*) – number of results that should be skipped
- **fields** (*Union[list, tuple, set], optional*) – names of fields (attributes) that should be included in results
- **search\_filters** (*Dict[str, Union[str, int, float]], optional*) – search filter criteria as key-value pairs, where *key* is a keyword or a tag of the attribute and *value* is the expected value that should match

**Returns** instance representations (see [Instance Result Attributes](#))

**Return type** List[Dict[str, dict]]

---

**Note:** The server may only return a subset of search results. In this case, a warning will notify the client that there are remaining results. Remaining results can be requested via repeated calls using the *offset* parameter.

---

**search\_for\_series** (*study\_instance\_uid=None, fuzzymatching=None, limit=None, offset=None, fields=None, search\_filters=None*)

Searches for DICOM series.

**Parameters**

- **study\_instance\_uid** (*str, optional*) – unique study identifier
- **fuzzymatching** (*bool, optional*) – whether fuzzy semantic matching should be performed
- **limit** (*int, optional*) – maximum number of results that should be returned
- **offset** (*int, optional*) – number of results that should be skipped
- **fields** (*Union[list, tuple, set], optional*) – names of fields (attributes) that should be included in results
- **search\_filters** (*Dict[str, Union[str, int, float]], optional*) – search filter criteria as key-value pairs, where *key* is a keyword or a tag of the attribute and *value* is the expected value that should match

**Returns** series representations (see [Series Result Attributes](#))

**Return type** List[Dict[str, dict]]

---

**Note:** The server may only return a subset of search results. In this case, a warning will notify the client that there are remaining results. Remaining results can be requested via repeated calls using the *offset* parameter.

---

**search\_for\_studies** (*fuzzymatching=None*, *limit=None*, *offset=None*, *fields=None*,  
*search\_filters=None*)

Searches for DICOM studies.

#### Parameters

- **fuzzymatching** (*bool*, *optional*) – whether fuzzy semantic matching should be performed
- **limit** (*int*, *optional*) – maximum number of results that should be returned
- **offset** (*int*, *optional*) – number of results that should be skipped
- **fields** (*Sequence[str]*, *optional*) – names of fields (attributes) that should be included in results
- **search\_filters** (*dict*, *optional*) – search filter criteria as key-value pairs, where *key* is a keyword or a tag of the attribute and *value* is the expected value that should match

**Returns** study representations (see [Study Result Attributes](#))

**Return type** List[Dict[str, dict]]

---

**Note:** The server may only return a subset of search results. In this case, a warning will notify the client that there are remaining results. Remaining results can be requested via repeated calls using the *offset* parameter.

---

**set\_http\_retry\_params** (*retry=True*, *max\_attempts=5*, *wait\_exponential\_multiplier=1000*,  
*retriable\_error\_codes=(<HTTPStatus.TOO\_MANY\_REQUESTS: 429>*, *<HTTPStatus.REQUEST\_TIMEOUT: 408>*, *<HTTP-*  
*Status.SERVICE\_UNAVAILABLE: 503>*, *<HTTPSta-*  
*tus.GATEWAY\_TIMEOUT: 504>)*)

Sets parameters for HTTP retrying logic. These parameters are passed to `@retrying.retry` which wraps the HTTP requests and retries all responses that return an error code defined in **retriable\_error\_codes**. The retrying method uses exponential back off using the multiplier **wait\_exponential\_multiplier** for a max attempts defined by **max\_attempts**.

#### Parameters

- **retry** (*bool*, *optional*) – whether HTTP retrying should be performed, if it is set to `False`, the rest of the parameters are ignored.
- **max\_attempts** (*int*, *optional*) – the maximum number of request attempts.
- **wait\_exponential\_multiplier** (*float*, *optional*) – exponential multiplier applied to delay between attempts in ms.
- **retriable\_error\_codes** (*tuple*, *optional*) – tuple of HTTP error codes to retry if raised.

**Return type** None



**store\_instances** (*datasets*, *study\_instance\_uid=None*)

Stores DICOM instances.

**Parameters**

- **datasets** (*Sequence* [*pydicom.dataset.Dataset*]) – instances that should be stored
- **study\_instance\_uid** (*str*, *optional*) – unique study identifier

**Returns** information about status of stored instances

**Return type** *pydicom.dataset.Dataset*

`dicomweb_client.api.load_json_dataset` (*dataset*)

Loads DICOM Data Set in DICOM JSON format.

**Parameters** **dataset** (*Dict* [*str*, *dict*]) – mapping where keys are DICOM *Tags* and values are mappings of DICOM *VR* and *Value* key-value pairs

**Returns** data set

**Return type** *pydicom.dataset.Dataset*

## 7.1.2 dicomweb\_client.cli module

Command Line Interface (CLI)

`dicomweb_client.cli.main` (*args*)

Main entry point for the `dicomweb_client` command line program.

### dicomweb\_client

Client for DICOMweb RESTful services.

```
usage: dicomweb_client [-h] [-v] [-u NAME] [-p PASSWORD] [--ca CERT-FILE]
                    [--cert CERT-FILE] [--bearer-token TOKEN] [--url URL]
                    [--chunk-size NUM]
                    {search,retrieve,store} ...
```

**-h, --help**

show this help message and exit

**-v, --verbosity**

logging verbosity that maps to a logging level (default: error, -v: warning, -vv: info, -vvv: debug, -vvvv: debug + traceback); all log messages are written to standard error

**-u <name>, --user <name>**

username for authentication with the DICOMweb service

**-p <password>, --password <password>**

password for authentication with the DICOMweb service

**--ca <cert-file>**

path to a CA bundle file

**--cert <cert-file>**

path to a client certificate file in PEM format

**--bearer-token <token>**

bearer token for authentication with the DICOMweb service

- url** <url>  
uniform resource locator of the DICOMweb service
- chunk-size** <num>  
maximum size of a network transfer chunk in bytes

### dicomweb\_client retrieve

WADO-RS: Web Access to DICOM Objects by RESTful Services.

```
usage: dicomweb_client retrieve [-h] INFORMATION ENTITIES ...
```

- h, --help**  
show this help message and exit

### dicomweb\_client retrieve bulkdata

Retrieve bulk data of a DICOM object from a known location.

```
usage: dicomweb_client retrieve bulkdata [-h]
                                     [--media-type MEDIATYPE [MEDIATYPE ...]]
                                     --uri URI
```

- h, --help**  
show this help message and exit
- media-type** <mediatype>  
acceptable media type and the optionally the UID of a corresponding transfer syntax separated by a white-space (e.g., “image/jpeg” or “image/jpeg 1.2.840.10008.1.2.4.50”)
- uri** <uri>  
unique resource identifier of bulk data element

### dicomweb\_client retrieve instances

Retrieve data for an individual DICOM instance.

```
usage: dicomweb_client retrieve instances [-h] --study UID --series UID
                                     --instance UID
                                     {metadata,full,frames} ...
```

- h, --help**  
show this help message and exit
- study** <uid>  
unique study identifier (StudyInstanceUID)
- series** <uid>  
unique series identifier (SeriesInstanceUID)
- instance** <uid>  
unique instance identifier (SOPInstanceUID)

## dicomweb\_client retrieve instances frames

Retrieve one or more frames of the pixel data element of an individual DICOM instance.

```
usage: dicomweb_client retrieve instances frames [-h] [--save]
                                                [--output-dir PATH]
                                                [--media-type MEDIATYPE [MEDIATYPE ..
↔.]]
                                                [--numbers NUM [NUM ...]]
                                                [--show]
```

**-h, --help**

show this help message and exit

**--save**

whether downloaded data should be saved

**--output-dir <path>**

path to directory where downloaded data should be saved

**--media-type <mediatype>**

acceptable media type and the optionally the UID of a corresponding transfer syntax separated by a whitespace (e.g., “image/jpeg” or “image/jpeg 1.2.840.10008.1.2.4.50”)

**--numbers <num>**

frame numbers

**--show**

display retrieved images

## dicomweb\_client retrieve instances full

Retrieve a DICOM instance.

```
usage: dicomweb_client retrieve instances full [-h] [--save]
                                                [--output-dir PATH]
                                                [--media-type MEDIATYPE [MEDIATYPE ...
↔]]
```

**-h, --help**

show this help message and exit

**--save**

whether downloaded data should be saved

**--output-dir <path>**

path to directory where downloaded data should be saved

**--media-type <mediatype>**

acceptable media type and the optionally the UID of a corresponding transfer syntax separated by a whitespace (e.g., “image/jpeg” or “image/jpeg 1.2.840.10008.1.2.4.50”)

## dicomweb\_client retrieve instances metadata

Retrieve metadata of an individual DICOM instance.

```
usage: dicomweb_client retrieve instances metadata [-h]
                                                [--prettyify | --dicomize]
                                                [--save]
                                                [--output-dir PATH]
```

- h, --help**  
show this help message and exit
- prettyify**  
pretty print JSON response message
- dicomize**  
convert JSON response message to DICOM data set
- save**  
whether downloaded data should be saved
- output-dir <path>**  
path to directory where downloaded data should be saved

### dicomweb\_client retrieve series

Retrieve data for all DICOM instances of a given DICOM series.

```
usage: dicomweb_client retrieve series [-h] --study UID --series UID
                                       {metadata,full} ...
```

- h, --help**  
show this help message and exit
- study <uid>**  
unique study identifier (StudyInstanceUID)
- series <uid>**  
unique series identifier (SeriesInstanceUID)

### dicomweb\_client retrieve series full

Retrieve DICOM instances of a given DICOM series.

```
usage: dicomweb_client retrieve series full [-h] [--save] [--output-dir PATH]
                                             [--media-type MEDIATYPE [MEDIATYPE ...]]
```

- h, --help**  
show this help message and exit
- save**  
whether downloaded data should be saved
- output-dir <path>**  
path to directory where downloaded data should be saved
- media-type <mediatype>**  
acceptable media type and the optionally the UID of a corresponding transfer syntax separated by a whitespace (e.g., “image/jpeg” or “image/jpeg 1.2.840.10008.1.2.4.50”)

### dicomweb\_client retrieve series metadata

Retrieve metadata of DICOM instances of a given DICOM series.

```
usage: dicomweb_client retrieve series metadata [-h] [--prettyfy | --dicomize]
                                               [--save] [--output-dir PATH]
```

- h, --help**  
show this help message and exit
- prettyfy**  
pretty print JSON response message
- dicomize**  
convert JSON response message to DICOM data set
- save**  
whether downloaded data should be saved
- output-dir <path>**  
path to directory where downloaded data should be saved

### dicomweb\_client retrieve studies

Retrieve data for all DICOM instances of a given DICOM study.

```
usage: dicomweb_client retrieve studies [-h] --study UID {metadata,full} ...
```

- h, --help**  
show this help message and exit
- study <uid>**  
unique study identifier (StudyInstanceUID)

### dicomweb\_client retrieve studies full

Retrieve DICOM instances of a given DICOM study.

```
usage: dicomweb_client retrieve studies full [-h] [--save] [--output-dir PATH]
                                             [--media-type MEDIATYPE [MEDIATYPE ...]]
```

- h, --help**  
show this help message and exit
- save**  
whether downloaded data should be saved
- output-dir <path>**  
path to directory where downloaded data should be saved
- media-type <mediatype>**  
acceptable media type and the optionally the UID of a corresponding transfer syntax separated by a white-space(e.g., “image/jpeg” or “image/jpeg 1.2.840.10008.1.2.4.50”)

### dicomweb\_client retrieve studies metadata

Retrieve metadata of DICOM instances of a given DICOM study.

```
usage: dicomweb_client retrieve studies metadata [-h]
                                                [--prettyify | --dicomize]
                                                [--save] [--output-dir PATH]
```

- h, --help**  
show this help message and exit
- prettyify**  
pretty print JSON response message
- dicomize**  
convert JSON response message to DICOM data set
- save**  
whether downloaded data should be saved
- output-dir <path>**  
path to directory where downloaded data should be saved

### dicomweb\_client search

QIDO-RS: Query based on ID for DICOM Objects by RESTful Serices.

```
usage: dicomweb_client search [-h] INFORMATION ENTITIES ...
```

- h, --help**  
show this help message and exit

### dicomweb\_client search instances

Search for DICOM instances.

```
usage: dicomweb_client search instances [-h] [--prettyify | --dicomize]
                                         [--filter KEY=VALUE] [--field NAME]
                                         [--limit NUM] [--offset NUM] [--fuzzy]
                                         [--study UID] [--series UID]
```

- h, --help**  
show this help message and exit
- prettyify**  
pretty print JSON response message
- dicomize**  
convert JSON response message to DICOM data set
- filter <key=value>**  
query filter criterion
- field <name>**  
field that should be included in response

- limit** <num>  
number of items that should be maximally retrieved
- offset** <num>  
number of items that should be skipped
- fuzzy**  
perform fuzzy matching
- study** <uid>  
unique study identifier (StudyInstanceUID)
- series** <uid>  
unique series identifier (SeriesInstanceUID)

### dicomweb\_client search series

Search for DICOM series.

```
usage: dicomweb_client search series [-h] [--filter KEY=VALUE] [--field NAME]
                                     [--limit NUM] [--offset NUM] [--fuzzy]
                                     [--prettify | --dicomize] [--study UID]
```

- h, --help**  
show this help message and exit
- filter** <key=value>  
query filter criterion
- field** <name>  
field that should be included in response
- limit** <num>  
number of items that should be maximally retrieved
- offset** <num>  
number of items that should be skipped
- fuzzy**  
perform fuzzy matching
- prettify**  
pretty print JSON response message
- dicomize**  
convert JSON response message to DICOM data set
- study** <uid>  
unique study identifier (StudyInstanceUID)

### dicomweb\_client search studies

Search for DICOM studies.

```
usage: dicomweb_client search studies [-h] [--filter KEY=VALUE] [--field NAME]
                                       [--limit NUM] [--offset NUM] [--fuzzy]
                                       [--prettify | --dicomize]
```

**-h, --help**  
show this help message and exit

**--filter** <key=value>  
query filter criterion

**--field** <name>  
field that should be included in response

**--limit** <num>  
number of items that should be maximally retrieved

**--offset** <num>  
number of items that should be skipped

**--fuzzy**  
perform fuzzy matching

**--prettify**  
pretty print JSON response message

**--dicomize**  
convert JSON response message to DICOM data set

### dicomweb\_client store

STOW-RS: Store Over the Web by RESTful Services.

```
usage: dicomweb_client store [-h] INFORMATION ENTITIES ...
```

**-h, --help**  
show this help message and exit

### dicomweb\_client store instances

Store DICOM instances.

```
usage: dicomweb_client store instances [-h] [--study UID] PATH [PATH ...]
```

**path**  
paths to DICOM files that should be loaded

**-h, --help**  
show this help message and exit

**--study** <uid>  
unique study identifier (StudyInstanceUID)

### 7.1.3 dicomweb\_client.error module

Custom error classes

**exception** `dicomweb_client.error.DICOMJSONError`  
Bases: `ValueError`  
Exception class for malformed DICOM JSON.



## 7.1.4 dicomweb\_client.log module

Utility functions for logging configuration

`dicomweb_client.log.configure_logging` (*verbosity*)

Configures the root logger with a “stderr” stream handler that directs logging messages to standard error (to allow capturing program standard output, e.g. in order to redirect it to a file).

Logging verbosity maps to levels as follows:

```
0 -> no messages
1 -> CRITICAL, ERROR & WARN/WARNING messages
2 -> CRITICAL, ERROR, WARN/WARNING, & INFO messages
3 -> CRITICAL, ERROR, WARN/WARNING, INFO & DEBUG messages
4 -> all messages
```

**Parameters** `verbosity` (*int*) – logging verbosity

**Returns** package root logger

**Return type** `logging.Logger`

## 7.1.5 dicomweb\_client.session\_utils module

`dicomweb_client.session_utils.add_certs_to_session` (*session*, *ca\_bundle=None*, *cert=None*)

Adds CA bundle and certificate to an existing session.

**Parameters**

- **session** (*requests.Session*) – input session
- **ca\_bundle** (*str, optional*) – path to CA bundle file
- **cert** (*str, optional*) – path to client certificate file in Privacy Enhanced Mail (PEM) format

**Returns** verified session

**Return type** `requests.Session`

`dicomweb_client.session_utils.create_session` ()

Creates an unauthorized session.

**Returns** unauthorized session

**Return type** `requests.Session`

`dicomweb_client.session_utils.create_session_from_auth` (*auth*)

Creates a session from a given AuthBase object.

**Parameters** **auth** (*requests.auth.AuthBase*) – an implementation of *requests.auth.AuthBase* to be used for authentication with services

**Returns** authorized session

**Return type** `requests.Session`

`dicomweb_client.session_utils.create_session_from_gcp_credentials` (*google\_credentials=None*)

Creates an authorized session for Google Cloud Platform.

**Parameters** `google_credentials` (*Any*) – Google cloud credentials. (see <https://cloud.google.com/docs/authentication/production> for more information on Google cloud authentication). If not set, will be initialized to `google.auth.default()`

**Returns** Google cloud authorized session

**Return type** `requests.Session`

`dicomweb_client.session_utils.create_session_from_user_pass` (*username*, *password*)

Creates a session from a given username and password.

**Parameters**

- **username** (*str*) – username for authentication with services
- **password** (*str*) – password for authentication with services

**Returns** authorized session

**Return type** `requests.Session`

## CHAPTER 8

---

### Indices and tables

---

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



**d**

`dicomweb_client.api`, 21  
`dicomweb_client.cli`, 29  
`dicomweb_client.error`, 36  
`dicomweb_client.log`, 37  
`dicomweb_client.session_utils`, 37



## Symbols

- bearer-token <token>
  - dicomweb\_client command line option, 29
- ca <cert-file>
  - dicomweb\_client command line option, 29
- cert <cert-file>
  - dicomweb\_client command line option, 29
- chunk-size <num>
  - dicomweb\_client command line option, 30
- dicomize
  - dicomweb\_client-retrieve-instances-metadata command line option, 32
  - dicomweb\_client-retrieve-series-metadata command line option, 33
  - dicomweb\_client-retrieve-studies-metadata command line option, 34
  - dicomweb\_client-search-instances command line option, 34
  - dicomweb\_client-search-series command line option, 35
  - dicomweb\_client-search-studies command line option, 36
- field <name>
  - dicomweb\_client-search-instances command line option, 34
  - dicomweb\_client-search-series command line option, 35
  - dicomweb\_client-search-studies command line option, 36
- filter <key=value>
  - dicomweb\_client-search-instances command line option, 34
  - dicomweb\_client-search-series command line option, 35
  - dicomweb\_client-search-studies command line option, 36
- command line option, 36
- fuzzy
  - dicomweb\_client-search-instances command line option, 35
  - dicomweb\_client-search-series command line option, 35
  - dicomweb\_client-search-studies command line option, 36
- instance <uid>
  - dicomweb\_client-retrieve-instances command line option, 30
- limit <num>
  - dicomweb\_client-search-instances command line option, 34
  - dicomweb\_client-search-series command line option, 35
  - dicomweb\_client-search-studies command line option, 36
- media-type <mediatype>
  - dicomweb\_client-retrieve-bulkdata command line option, 30
  - dicomweb\_client-retrieve-instances-frames command line option, 31
  - dicomweb\_client-retrieve-instances-full command line option, 31
  - dicomweb\_client-retrieve-series-full command line option, 32
  - dicomweb\_client-retrieve-studies-full command line option, 33
- numbers <num>
  - dicomweb\_client-retrieve-instances-frames command line option, 31
- offset <num>
  - dicomweb\_client-search-instances command line option, 35
  - dicomweb\_client-search-series command line option, 35
  - dicomweb\_client-search-studies command line option, 36
- output-dir <path>

```

dicomweb_client-retrieve-instances-frames  command line option,30
  command line option,31
dicomweb_client-retrieve-instances-full    command line option,32
  command line option,31
dicomweb_client-retrieve-instances-metadata command line option,33
  command line option,32
dicomweb_client-retrieve-series-full      command line option,35
  command line option,32
dicomweb_client-retrieve-series-metadata  command line option,35
  command line option,33
dicomweb_client-retrieve-studies-full     command line option,36
  command line option,33
dicomweb_client-retrieve-studies-metadata dicomweb_client-retrieve-bulkdata
  command line option,34                          command line option,30
-prettyfy                                  -url <url>
dicomweb_client-retrieve-instances-metadata dicomweb_client command line
  command line option,32                          option,29
dicomweb_client-retrieve-series-metadata  -help
  command line option,33                          dicomweb_client command line
dicomweb_client-retrieve-studies-metadata option,29
  command line option,34                          dicomweb_client-retrieve command
dicomweb_client-search-instances          line option,30
  command line option,34                          dicomweb_client-retrieve-bulkdata
dicomweb_client-search-series             command line option,30
  command line option,35                          dicomweb_client-retrieve-instances
dicomweb_client-search-studies            command line option,30
  command line option,36                          dicomweb_client-retrieve-instances-frames
-save                                       command line option,31
dicomweb_client-retrieve-instances-frames dicomweb_client-retrieve-instances-full
  command line option,31                          command line option,31
dicomweb_client-retrieve-instances-full  dicomweb_client-retrieve-instances-metadata
  command line option,31                          command line option,32
dicomweb_client-retrieve-instances-metadata dicomweb_client-retrieve-series
  command line option,32                          command line option,32
dicomweb_client-retrieve-series-full     dicomweb_client-retrieve-series-full
  command line option,32                          command line option,32
dicomweb_client-retrieve-series-metadata dicomweb_client-retrieve-series-metadata
  command line option,33                          command line option,33
dicomweb_client-retrieve-studies-full    dicomweb_client-retrieve-studies
  command line option,33                          command line option,33
dicomweb_client-retrieve-studies-metadata dicomweb_client-retrieve-studies-full
  command line option,34                          command line option,33
-series <uid>                              dicomweb_client-retrieve-studies-metadata
  dicomweb_client-retrieve-instances          command line option,34
  command line option,30                      dicomweb_client-search command
  dicomweb_client-retrieve-series            line option,34
  command line option,32                      dicomweb_client-search-instances
  dicomweb_client-search-instances          command line option,34
  command line option,35                      dicomweb_client-search-series
-show                                       command line option,35
  dicomweb_client-retrieve-instances-frames dicomweb_client-search-studies
  command line option,31                          command line option,35
-study <uid>                               dicomweb_client-store command line
  dicomweb_client-retrieve-instances          option,36

```



dicomweb\_client-store-instances  
 command line option, 36

-p <password>, -password <password>  
 dicomweb\_client command line  
 option, 29

-u <name>, -user <name>  
 dicomweb\_client command line  
 option, 29

-v, -verbosity  
 dicomweb\_client command line  
 option, 29

## A

add\_certs\_to\_session() (in module *dicomweb\_client.session\_utils*), 37

## B

base\_url (*dicomweb\_client.api.DICOMwebClient* attribute), 21

## C

configure\_logging() (in module *dicomweb\_client.log*), 37

create\_session() (in module *dicomweb\_client.session\_utils*), 37

create\_session\_from\_auth() (in module *dicomweb\_client.session\_utils*), 37

create\_session\_from\_gcp\_credentials()  
 (in module *dicomweb\_client.session\_utils*), 37

create\_session\_from\_user\_pass() (in module *dicomweb\_client.session\_utils*), 38

## D

delete\_instance() (*dicomweb\_client.api.DICOMwebClient* method), 22

delete\_series() (*dicomweb\_client.api.DICOMwebClient* method), 23

delete\_study() (*dicomweb\_client.api.DICOMwebClient* method), 23

delete\_url\_prefix (*dicomweb\_client.api.DICOMwebClient* attribute), 22

DICOMJSONError, 36

dicomweb\_client command line option

- bearer-token <token>, 29
- ca <cert-file>, 29
- cert <cert-file>, 29
- chunk-size <num>, 30
- url <url>, 29
- h, -help, 29

- p <password>, -password <password>, 29

- u <name>, -user <name>, 29
- v, -verbosity, 29

dicomweb\_client-retrieve command line option

- h, -help, 30

dicomweb\_client-retrieve-bulkdata  
 command line option

- media-type <mediatype>, 30

- uri <uri>, 30

- h, -help, 30

dicomweb\_client-retrieve-instances  
 command line option

- instance <uid>, 30

- series <uid>, 30

- study <uid>, 30

- h, -help, 30

dicomweb\_client-retrieve-instances-frames  
 command line option

- media-type <mediatype>, 31

- numbers <num>, 31

- output-dir <path>, 31

- save, 31

- show, 31

- h, -help, 31

dicomweb\_client-retrieve-instances-full  
 command line option

- media-type <mediatype>, 31

- output-dir <path>, 31

- save, 31

- h, -help, 31

dicomweb\_client-retrieve-instances-metadata  
 command line option

- dicomize, 32

- output-dir <path>, 32

- prettify, 32

- save, 32

- h, -help, 32

dicomweb\_client-retrieve-series  
 command line option

- series <uid>, 32

- study <uid>, 32

- h, -help, 32

dicomweb\_client-retrieve-series-full  
 command line option

- media-type <mediatype>, 32

- output-dir <path>, 32

- save, 32

- h, -help, 32

dicomweb\_client-retrieve-series-metadata  
 command line option

- dicomize, 33

- output-dir <path>, 33

-prettify, 33  
 -save, 33  
 -h, -help, 33  
 dicomweb\_client-retrieve-studies  
     command line option  
     -study <uid>, 33  
     -h, -help, 33  
 dicomweb\_client-retrieve-studies-full  
     command line option  
     -media-type <mediatype>, 33  
     -output-dir <path>, 33  
     -save, 33  
     -h, -help, 33  
 dicomweb\_client-retrieve-studies-metadata  
     command line option  
     -dicomize, 34  
     -output-dir <path>, 34  
     -prettify, 34  
     -save, 34  
     -h, -help, 34  
 dicomweb\_client-search command line  
     option  
     -h, -help, 34  
 dicomweb\_client-search-instances  
     command line option  
     -dicomize, 34  
     -field <name>, 34  
     -filter <key=value>, 34  
     -fuzzy, 35  
     -limit <num>, 34  
     -offset <num>, 35  
     -prettify, 34  
     -series <uid>, 35  
     -study <uid>, 35  
     -h, -help, 34  
 dicomweb\_client-search-series command  
     line option  
     -dicomize, 35  
     -field <name>, 35  
     -filter <key=value>, 35  
     -fuzzy, 35  
     -limit <num>, 35  
     -offset <num>, 35  
     -prettify, 35  
     -study <uid>, 35  
     -h, -help, 35  
 dicomweb\_client-search-studies command  
     line option  
     -dicomize, 36  
     -field <name>, 36  
     -filter <key=value>, 36  
     -fuzzy, 36  
     -limit <num>, 36  
     -offset <num>, 36

-prettify, 36  
 -h, -help, 35  
 dicomweb\_client-store command line  
     option  
     -h, -help, 36  
 dicomweb\_client-store-instances  
     command line option  
     -study <uid>, 36  
     -h, -help, 36  
     path, 36  
 dicomweb\_client.api (module), 21  
 dicomweb\_client.cli (module), 29  
 dicomweb\_client.error (module), 36  
 dicomweb\_client.log (module), 37  
 dicomweb\_client.session\_utils (module), 37  
 DICOMwebClient (class in dicomweb\_client.api), 21

## H

host (dicomweb\_client.api.DICOMwebClient attribute), 21

## L

load\_json\_dataset() (in module dicomweb\_client.api), 29  
 lookup\_keyword() (dicomweb\_client.api.DICOMwebClient static method), 23  
 lookup\_tag() (dicomweb\_client.api.DICOMwebClient static method), 23

## M

main() (in module dicomweb\_client.cli), 29

## P

path  
     dicomweb\_client-store-instances  
         command line option, 36  
 port (dicomweb\_client.api.DICOMwebClient attribute), 21  
 protocol (dicomweb\_client.api.DICOMwebClient attribute), 21

## Q

qido\_url\_prefix (dicomweb\_client.api.DICOMwebClient attribute), 22

## R

retrieve\_bulkdata() (dicomweb\_client.api.DICOMwebClient method), 24  
 retrieve\_instance() (dicomweb\_client.api.DICOMwebClient method), 24

`retrieve_instance_frames()` (*dicomweb\_client.api.DICOMwebClient* method), 24  
`retrieve_instance_frames_rendered()` (*dicomweb\_client.api.DICOMwebClient* method), 24  
`retrieve_instance_metadata()` (*dicomweb\_client.api.DICOMwebClient* method), 25  
`retrieve_instance_rendered()` (*dicomweb\_client.api.DICOMwebClient* method), 25  
`retrieve_series()` (*dicomweb\_client.api.DICOMwebClient* method), 26  
`retrieve_series_metadata()` (*dicomweb\_client.api.DICOMwebClient* method), 26  
`retrieve_series_rendered()` (*dicomweb\_client.api.DICOMwebClient* method), 26  
`retrieve_study()` (*dicomweb\_client.api.DICOMwebClient* method), 26  
`retrieve_study_metadata()` (*dicomweb\_client.api.DICOMwebClient* method), 26

## S

`search_for_instances()` (*dicomweb\_client.api.DICOMwebClient* method), 27  
`search_for_series()` (*dicomweb\_client.api.DICOMwebClient* method), 27  
`search_for_studies()` (*dicomweb\_client.api.DICOMwebClient* method), 28  
`set_http_retry_params()` (*dicomweb\_client.api.DICOMwebClient* method), 28  
`store_instances()` (*dicomweb\_client.api.DICOMwebClient* method), 28  
`stow_url_prefix` (*dicomweb\_client.api.DICOMwebClient* attribute), 22

## U

`url_prefix` (*dicomweb\_client.api.DICOMwebClient* attribute), 21

## W

`wado_url_prefix` (*dicomweb\_client.api.DICOMwebClient* attribute), 21