

**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

## About this Tutorial

This tutorial is necessary to retrieve the dataset for participating in the [SIIM-ACR Pneumothorax Segmentation Competition](#) on Kaggle. The dataset is only hosted on Google Cloud Platform (GCP) through the Cloud Healthcare (CHC) API.

There are methods to retrieve these datasets at no cost to you. And it is also possible to use GCP to do your modeling, with associated pricing structures for those activities, depending on which tools are used. This competition is also issuing GCP Credits for users interested in exploring the latter. Review the competition's Discussion Forum for more details.

## Tutorial Menu

[About this Tutorial](#)

[Tutorial Menu](#)

[Prerequisites](#)

[About the Cloud Healthcare API](#)

[Authentication](#)

[Introduction to DICOMWeb](#)

[Exploring and Downloading the DICOM instances](#)

[Method 1: Downloading DICOM Studies, Series, Instances or Frames Locally](#)

[Method 2: Export DICOM instances to your GCS bucket](#)

[Downloading the FHIR annotations](#)

[Other Cloud Healthcare API Actions](#)

## Prerequisites

- A Google Cloud account. If you don't have an account, [register for one here](#). For new users, the [Google Cloud Products \(GCP\) free tier](#) will give you credits to use on GCP for things like storage or using GCP ML tools where there are costs incurred.
- A [Google Cloud project](#).
- The google email associated with your Google Cloud account must [join the SIIM-Kaggle google group](#) to gain permissions to the necessary Healthcare dataset. Note that once you've completed the "join" to the group, you will self-join without any further approval. There is no further action to take. It is not a real e-mail group, so you should not see any posts nor

**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

should you be able to make any posts. It exists purely to provision access to the competition's datastore.

- [Download and install gcloud.](#)
- [Download and install gsutil.](#)

## About the Cloud Healthcare API

The [Cloud Healthcare API](#) is a managed service that can be used to store, retrieve, and query medical formats such as DICOM and FHIR. In the following steps, we will use this API to query and retrieve the DICOM images and the images' FHIR annotations.

## Authentication

The easiest way to authenticate is to use Application Default Credentials. This will set up all Cloud SDK tooling and API client libraries to use your user credentials.

In a new shell, run the following command and then follow the instructions:

```
gcloud auth application-default login
```

**Note:** There are various other ways to authenticate to the Cloud Healthcare API. The method you choose depends on many factors, including whether you are running as an end-user or as a service account (useful for running in a GCP VM). The following guides discuss the authentication options:

- [Authorizing Cloud SDK tools](#)
- [Authentication to the Cloud Healthcare API](#)

## Introduction to DICOMWeb

The [DICOMWeb protocol](#) can be used to interact with the Cloud Healthcare API's DICOMWeb endpoint. DICOM objects are structured in this form:



**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

Think of an instance as a single object, like an image or a report. The series can contain multiple instances (e.g. a CT scan series). The study can contain multiple series (e.g. a series with medical images, and a series with a report). In the dataset we are going to deal with most studies contain a single series, and most series contain a single instance.

In the “Exploring and Downloading DICOM instances” section below, you’ll be presented with 2 methods to retrieve the instances from the competition datasets.

In the “Downloading the FHIR annotations” section below, you’ll be shown how to retrieve the training set annotations.

## Exploring and Downloading the DICOM instances

The [DICOMWeb protocol](#) can be used to interact with the Cloud Healthcare API’s DICOMWeb endpoint. For information about the features supported by this endpoint, see the [DICOM conformance statement](#).

There are two DICOM stores, one for hosting training data and one for hosting test data.

Training	Test
<i>PROJECT_ID</i> =kaggle-siim-healthcare	<i>PROJECT_ID</i> =kaggle-siim-healthcare
<i>REGION</i> =us-central1	<i>REGION</i> =us-central1
<i>DATASET_ID</i> =siim-pneumothorax	<i>DATASET_ID</i> =siim-pneumothorax
<i>DICOM_STORE_ID</i> =dicom-images-train	<i>DICOM_STORE_ID</i> =dicom-images-test

There will be 2 methods presented in this tutorial for retrieving the DICOM images:

1. **Download DICOM Studies, Series, Instances or Frames locally.** This can be achieved using the script provided. Please give it a some time as it is downloading around ~10K images.
2. **Export instances to your GCS bucket.** This is recommended if you want to work with the dataset with some other GCP tooling (i.e. GCE/GCK, Cloud ML Engine, Colaboratory). From the bucket, you can also download from the bucket locally. Note: You can expect to incur charges for the storage of the dataset to your GCS bucket (and possibly other costs associated with using Cloud tools).

### Method 1: Downloading DICOM Studies, Series, Instances or Frames Locally

This method uses a script to download the datastore (hosted on the `kaggle-siim-healthcare` project) to your local directory.

**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

### 1. Authentication

As previously mentioned in the Authentication section please run the following to authenticate your user credentials.

```
gcloud auth application-default login
```

### 2. Install the necessary Python libraries.

For the script below you will need the [retrying library](#), and the [google auth library](#). And, if running on Python 2.7 you will need to install the [futures library](#).

```
pip install retrying
pip install google-auth
# Only if running on Python 2.7
pip install futures
```

### 3. Run the script below.

The script downloads all images in both DICOM stores and saves them in the directory where the script is run. The train set is about 1.5 GB total. The test set is about 180 MB total.

We've also provided this script on the competition's [Data Page](#), in a script called `download\_images.py`

If, instead, you're interested in retrieving specific studies, series, instances, you should review the information under the "Other Cloud Healthcare API Actions" section. You'll have to then revise the script to retrieve the specific data you are looking for.

Note: Due to the large number of images, this may take **10-15 minutes**.

```
"""Script to download all instances in a DICOM Store."""
import os
import posixpath
from concurrent import futures
from retrying import retry
import google.auth
from google.auth.transport.requests import AuthorizedSession

# URL of CHC (Cloud Healthcare) API
CHC_API_URL = 'https://healthcare.googleapis.com/v1beta1'

PROJECT_ID = 'kaggle-siim-healthcare'
```

**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

```
REGION = 'us-central1'
DATASET_ID = 'siim-pneumothorax'
TRAIN_DICOM_STORE_ID = 'dicom-images-train'
TEST_DICOM_STORE_ID = 'dicom-images-test'

@retry(wait_exponential_multiplier=1000, wait_exponential_max=10000)
def download_instance(dicom_web_url, dicom_store_id, study_uid, series_uid,
                    instance_uid, credentials):
    """Downloads a DICOM instance and saves it under the current folder."""
    instance_url = posixpath.join(dicom_web_url, 'studies', study_uid, 'series',
                                  series_uid, 'instances', instance_uid)
    authed_session = AuthorizedSession(credentials)
    response = authed_session.get(
        instance_url, headers={'Accept': 'application/dicom; transfer-syntax=*'})
    file_path = posixpath.join(dicom_store_id, study_uid, series_uid,
                               instance_uid)
    filename = '%s.dcm' % file_path
    if not os.path.exists(filename):
        os.makedirs(os.path.dirname(filename))
    with open(filename, 'wb') as f:
        f.write(response.content)

def download_all_instances(dicom_store_id, credentials):
    """Downloads all DICOM instances in the specified DICOM store."""
    # Get a list of all instances.
    dicom_web_url = posixpath.join(CHC_API_URL, 'projects', PROJECT_ID,
                                   'locations', REGION, 'datasets', DATASET_ID,
                                   'dicomStores', dicom_store_id, 'dicomWeb')
    qido_url = posixpath.join(dicom_web_url, 'instances')
    authed_session = AuthorizedSession(credentials)
    response = authed_session.get(qido_url, params={'limit': '15000'})
    if response.status_code != 200:
        print(response.text)
        return
    content = response.json()
    # DICOM Tag numbers
    study_instance_uid_tag = '0020000D'
    series_instance_uid_tag = '0020000E'
    sop_instance_uid_tag = '00080018'
    value_key = 'Value'
    with futures.ThreadPoolExecutor() as executor:
        future_to_study_uid = {}
        for instance in content:
            study_uid = instance[study_instance_uid_tag][value_key][0]
            series_uid = instance[series_instance_uid_tag][value_key][0]
            instance_uid = instance[sop_instance_uid_tag][value_key][0]
            future = executor.submit(download_instance, dicom_web_url, dicom_store_id,
                                    study_uid, series_uid, instance_uid, credentials)
```

**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

```
future_to_study_uid[future] = study_uid
processed_count = 0
for future in futures.as_completed(future_to_study_uid):
    try:
        future.result()
        processed_count += 1
        if not processed_count % 100 or processed_count == len(content):
            print('Processed instance %d out of %d' %
                  (processed_count, len(content)))
    except Exception as e:
        print('Failed to download a study. UID: %s \n exception: %s' %
              (future_to_study_uid[future], e))

def main(argv=None):
    credentials, _ = google.auth.default()
    print('Downloading all instances in %s DICOM store' % TRAIN_DICOM_STORE_ID)
    download_all_instances(TRAIN_DICOM_STORE_ID, credentials)
    print('Downloading all instances in %s DICOM store' % TEST_DICOM_STORE_ID)
    download_all_instances(TEST_DICOM_STORE_ID, credentials)

main()
```

## Method 2: Export DICOM instances to your GCS bucket

This method will post the datastore's DICOM instances (hosted on the `kaggle-siim-healthcare` project) to a GCS bucket that you name in your GCP project.

### 1. Creating a GCS bucket

Google Cloud Storage (GCS) is a storage system used to store and access objects on GCP. The Cloud Healthcare API can export all DICOM instances to a GCS bucket. To do this, we first create a GCS bucket and assign it to a **BUCKET** variable. For more detailed guidance, see [Creating storage buckets](#). **MY\_PROJECT** represents the project name for your GCP project where you'll be transferring the dataset.

```
MY_PROJECT="MY_PROJECT"
BUCKET="gs://${USER}-siim-bucket-dicom/"
gsutil mb -p ${MY_PROJECT} -c regional -l us-central1 -b on ${BUCKET}
```

### 2. Granting permissions to export to GCS

To grant permissions for the Cloud Healthcare API to write to the new bucket, follow the instructions for [Exporting data to Cloud Storage](#). IMPORTANT: The roles required should be set

**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

for this member: [service-38125765871@gcp-sa-healthcare.iam.gserviceaccount.com](mailto:service-38125765871@gcp-sa-healthcare.iam.gserviceaccount.com) -- Note that without allowing the Storage Object Admin role for this specific member, you will not be able to download the dataset to a GCS bucket.

### 3. Run export operation to copy DICOM store into GCS bucket

To run the export operation and inspect its status, follow the instructions in [Exporting DICOM instances](#). You can find a code sample to export the **training** dataset below. You can use the **DIRECTORY** variable to control which sub-directory to output the DICOM instances to in the bucket.

Note: Depending on the notation required by the method you are using to execute the below code, your uriPrefix may need to be adjusted. The goal is to name it to match the format: 'gs://YOUR\_BUCKET/train' where YOUR\_BUCKET is the bucket you created in the prior section.

```
PROJECT_ID="kaggle-siim-healthcare"
REGION="us-central1"
DATASET_ID="siim-pneumothorax"
DICOM_STORE_ID="dicom-images-train"
DIRECTORY="train"
curl -X POST \

  -H "Authorization: Bearer "$(gcloud auth print-access-token) \

  -H "Content-Type: application/json; charset=utf-8" \

  --data "{

    'gcsDestination': {

      'uriPrefix': "'${BUCKET}${DIRECTORY}'"

    }

  }"

"https://healthcare.googleapis.com/v1beta1/projects/${PROJECT_ID}/locations/${REGION}/datasets/${DATASET_ID}/dicomStores/${DICOM_STORE_ID}:export"
```

Similarly, run the following command to retrieve the DICOM test set.

```
DIRECTORY_TEST="test"
DICOM_STORE_ID_TEST="dicom-images-test"
curl -X POST \
```

**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

```
-H "Authorization: Bearer "$(gcloud auth print-access-token) \  
-H "Content-Type: application/json; charset=utf-8" \  
--data "{  
  'gcsDestination': {  
    'uriPrefix': '"${BUCKET}${DIRECTORY_TEST}"'  
  }  
}"  
"https://healthcare.googleapis.com/v1beta1/projects/${PROJECT_ID}/locations/${REGION}/datasets/${DATASET_ID}/dicomStores/${DICOM_STORE_ID_TEST}:export"
```

#### 4. Check operation status

Check status of the long-running operation by replacing *OPERATION\_ID* with the operation ID that is returned in the JSON response to the prior request.

```
curl -X GET \  
-H "Authorization: Bearer "$(gcloud auth print-access-token) \  
"https://healthcare.googleapis.com/v1beta1/projects/${PROJECT_ID}/locations/${REGION}/datasets/${DATASET_ID}/operations/OPERATION_ID"
```

To list the contents of the output bucket, run the following command.

```
gsutil ls ${BUCKET}${DIRECTORY}/**
```

#### 5. Copy GCS bucket to local file system

To copy the GCS bucket to your local file system, run the following command:

```
mkdir /tmp/siim-dicom  
gsutil -m cp -R ${BUCKET}${DIRECTORY} /tmp/siim-dicom/
```



**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

## Downloading the FHIR annotations

The annotations for this dataset are stored as FHIR resources (specifically as a [DocumentReference](#) resource). The Cloud Healthcare API can be used to query and download these FHIR resources. The features supported by this endpoint can be found in the [FHIR conformance statement](#).

There is one FHIR store containing the training annotations:

*PROJECT\_ID*=kaggle-siim-healthcare

*REGION*=us-central1

*DATASET\_ID*=siim-pneumothorax

*FHIR\_STORE\_ID*=fhir-masks-train

*DOCUMENT\_REFERENCE\_ID*=d70d8f3e-990a-4bc0-b11f-c87349f5d4eb

The [DocumentReference](#) resources contain a URL to the CSV file hosted on [siim.org](#) that contains the annotated masks.

The following code sample will access the FHIR datastore for the **training** dataset. See [Getting a FHIR resource](#) for samples in other languages. **Note that the response from the below code will point to a URL for the .CSV that contains the annotated masks for each ImageId.**

```
PROJECT_ID="kaggle-siim-healthcare"
REGION="us-central1"
DATASET_ID="siim-pneumothorax"
FHIR_STORE_ID="fhir-masks-train"
DOCUMENT_REFERENCE_ID="d70d8f3e-990a-4bc0-b11f-c87349f5d4eb"

curl -X GET \
-H "Authorization: Bearer "$(gcloud auth print-access-token) \
"https://healthcare.googleapis.com/v1beta1/projects/${PROJECT_ID}/locations
/${REGION}/datasets/${DATASET_ID}/fhirStores/${FHIR_STORE_ID}/fhir/Document
Reference/${DOCUMENT_REFERENCE_ID}"
```

## Other Cloud Healthcare API Actions

### Searching for DICOM Instances

The [SearchTransaction](#) protocol can be used to search for studies, series, or instances in a dicomStore. The following code sample searches for all instances below on the **training** dataset.

**NOTE:** CLOSED FOR FURTHER EDITS & [CONVERTED TO .PDF](#) FOR THE COMPETITION on 6/23.

See [Searching for studies, series, instances, and frames](#) for additional examples. See [the supported search modes and the supported DICOM tags](#).

```
curl -X GET \  
-H "Authorization: Bearer "$(gcloud auth print-access-token) \  
"https://healthcare.googleapis.com/v1beta1/projects/${PROJECT_ID}/locations/  
/${REGION}/datasets/${DATASET_ID}/dicomStores/${DICOM_STORE_ID}/dicomWeb/in  
stances"
```

## Fetching a specific study, series, instance or frame

If you'd like to fetch specific study, series, instance or frame to your local machine, you can use the [RetrieveTransaction](#) protocol. You can find sample call below to retrieve an individual instance. For this example, a known Study UID from the **training** dataset has been declared. See [Retrieving a study, series, instance, or frame](#) for samples in other languages.

```
STUDY_UID="1.2.276.0.7230010.3.1.2.8323329.12562.1517875239.738011"  
  
curl -X GET \  
-H "Authorization: Bearer "$(gcloud auth print-access-token) \  
"https://healthcare.googleapis.com/v1beta1/projects/${PROJECT_ID}/locations/  
/${REGION}/datasets/${DATASET_ID}/dicomStores/${DICOM_STORE_ID}/dicomWeb/st  
udies/${STUDY_UID}"
```

The response of this call is [MIME Multipart](#). We can decode the MIME Multipart message and retrieve the individual instances using common libraries, such as <https://github.com/requests/toolbelt> (you will need to install this).

## Exporting DICOM metadata to BigQuery

You can export the metadata of all DICOMs in a DICOM store to BigQuery. This might be useful if you'd like to do some quick analytics on the metadata. For more information, please see <https://cloud.google.com/healthcare/docs/how-tos/dicom-export-bigquery>.