

V3 Storage

Introduction

V3 Storage is the image-oriented storage back-end for DG's cloud. This application manages individual files uploaded to the cloud, including:

- image attributes
- image PHI
- image transcoding (DCM --> JPEG) as well as study-level items like:
- Study Schema
- Transfer syntaxes
- Study PHI
- Study Attributes

Storage API

Study Notifications

Create Per-Study Notification

Request:

```
curl -v http://localhost:9080/internal-api/v2/study/notification/{name_of_notification} -XPUT -d @params.json -H "Content-type: application/json"
```

where params.json contains:

```
{
  "request_method": "GET",
  "request_url": "http://V3_SERVICE_IP_OR_DN:PORT/study/notify?storage_namespace={namespace}&study_uid={studyUid}&phi_namespace={phiNamespace}&callback_url=http://V3_STORAGE_IP_OR_DN:PORT&action={event_type}",
  "request_body": ""
}
```

action can be one of the following:

```
HARVEST
UPLOAD
STUDY_DELETE
PHI
ATTACHMENT
NONE
```

actor is the ID of the actor making the action.

Response:

```
HTTP 200/OK
```

When implementing the notification receiver service, ensure that the status from the service is HTTP 200/OK

Delete Per-Study Notifications

```
curl -v http://localhost:9080/internal-api/v2/study/notification/{name_of_notification} -XDELETE
```

Get List of Per-Study Notifications

```
curl -v http://localhost:9080/internal-api/v2/study/notification -XGET
```

Create Scan Notification

Request:

```
curl -v http://localhost:9080/internal-api/v2/study/scan -XPOST -d @params.json -H "Content-type: application/json"
```

where params.json contains:

```
{
  "from_ts": 12345657,
  "to_ts": 999999,
  "namespace": ["ns1", "ns2", ...],
  "request_method": "GET",
  "request_url": "http://V3_SERVICE_IP_OR_DN:PORT/study/notify?storage_namespace={namespace}&
study_uid={studyUid}&phi_namespace={phiNamespace}&callback_url=http://V3_STORAGE_IP_OR_D
N:PORT&action={event_type}&actor={actor}",
  "request_body": ""
}
```

where:

from_ts - 1212121 - storage timestamp millis inclusive

to_ts - 123123123- storage timestamp millis exclusive

namespace - ["ns1", "ns2", ...], If missing, all namespaces are scanned

Response:

```
HTTP 202/Accepted
```

Uploading Studies to DICOM Grid

In order to upload studies to DICOM Grid, there are two methods you can utilize:

1. Use the DG Gateway to upload studies with DICOM protocol
2. Use DG EasyShare wizard to upload studies using a web browser

DG Gateway

To upload studies using the DG Gateway, you need to install the DG Gateway and set it up with your organization. Instructions will be provided in a separate document

DG EasyShare Wizard

To upload studies from a web browser, we recommend using our DG EasyShare wizard. This wizard will walk the user through selecting the CD/drive where the images are located, it will process the DICOM header information in the files, allow the user to select which study(s) to upload, and will upload and share those studies with your organization.

The url for the DG EasyShare wizard is:

[https://\[YOUR_HOST_NAME\].dicomgrid.com/share/\[YOUR_SHARE_CODE\]](https://[YOUR_HOST_NAME].dicomgrid.com/share/[YOUR_SHARE_CODE])

Where:

YOUR_HOST_NAME = your custom host name, like 'access' as in 'access.dicomgrid.com'

YOUR_SHARE_CODE = a self-assigned easy-to-remember name for your organization - like 'DICOMGrid' - should be one word

Accessing Studies

Storage includes a link-based API for accessing images using the study viewer. In order to use this API, please follow the below detailed steps:

1. set up an API account that has VIEW STUDY permissions in your organization
2. assign this account to all locations/groups that may have studies you would like to view via the API link
3. use the following URL to embed a link to the study viewer into your application:
`http://[YOUR_HOSTNAME].dicomgrid.com/login.html?login=[APIACCTLOGIN]&password=[APIACCTPASSWORD]&study=[STORAGENAMESPACE]/[STUDYUID]&phnamespace=[PHNAMESPACE]`

Where:

YOUR_HOST_NAME = your custom host name, like 'access' as in 'access.dicomgrid.com'

API_ACCT_LOGIN = login email of the account set up above in step #1

API_ACCT_PASSWORD = password for the account set up above in step #1

STORAGE_NAMESPACE = contained within the study notification

PHI_NAMESPACE = contained within the study notification

STUDY_UID = contained within the study notification

Working w/ V3 Storage

GET Study-level Data

Schema: Returns all images + attachments, broken down by series for a study

request:

```
curl -X GET
```

```
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/schema?phi_namespace={phi_namespace} -H "Accept: application/json" -H "Cookie: sid={sid}"
```

response:

```
{
  "namespace": "0785926b-9a28-4fcc-b0d9-e57488647403",
  "study_uid": "1.2.124.113532.10.236.100.70.20090407.180234.3203602",
  "series": [
    {
      "id": "1",
      "images": [
        {
          "id": "1.3.12.2.1107.5.1.4.24552.4.0.1052102647181713",
          "version": "04gnjBl3EVwL554SxOCr5gHCxW4",
          "rank": 1,
          "frame_count": 1
        },
        {
          "id": "1.3.12.2.1107.5.8.2.484849.838377.68674849.2009040803213532",
          "version": "3-2Rnh3gO5HNbq6l4R3cHiYIOF4",
          "rank": 2,
          "frame_count": 1
        }
      ]
    }
  ]
}
```

PHI: Returns "average" PHI for the study

request:

```
curl -X GET http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/phi -H "Accept: application/json" -H "Cookie: sid={sid}"
```

response:

```
{
  "tag": "1.2.124.113532.10.236.100.70.20090407.180234.3203602",
  "value": "1.2.124.113532.10.236.100.70.20090407.180234.3203602",
  "attr": [
    {
      "tag": "(0008,0050)",
      "value": "0000000"
    },
    {
      "tag": "(0008,1030)",
      "value": "STUDY^DESCRIPTION"
    }
  ]
}
```

```
{
  "tag": "(0010,0010)",
  "value": "PATIENT_NAME"
},
{
  "tag": "(0010,0020)",
  "value": "000000"
},
{
  "tag": "(0010,0030)",
  "value": "19000101"
},
{
  "tag": "(0010,0040)",
  "value": "F"
},
{
  "tag": "(0010,1010)",
  "value": "100Y"
},
{
  "tag": "(0010,1040)",
  "value": "PATIENT_ADDRESS"
}
]
}
```

PHI: Returns "average" PHI for the study using a different PHI namespace

request:

```
curl -X GET
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/phi?phi_namespace={phi_namespace} -H "Accept: application/json" -H "Cookie: sid={sid}"
```

response:

```
{
  "tag": "1.2.124.113532.10.236.100.70.20090407.180234.3203602",
  "value": "1.2.124.113532.10.236.100.70.20090407.180234.3203602",
  "attr": [
    {
      "tag": "(0008,0050)",
      "value": "0000000"
    },
    {
      "tag": "(0008,1030)",
      "value": "STUDY^DESCRIPTION"
    },
    {
      "tag": "(0010,0010)",
```

```

    "value":"PATIENT_NAME"
  },
  {
    "tag":"(0010,0020)",
    "value":"000000"
  },
  {
    "tag":"(0010,0030)",
    "value":"19000101"
  },
  {
    "tag":"(0010,0040)",
    "value":"F"
  },
  {
    "tag":"(0010,1010)",
    "value":"100Y"
  },
  {
    "tag":"(0010,1040)",
    "value":"PATIENT_ADDRESS"
  }
]
}

```

Tags: Returns the average of commonly referenced tags for the study (PHI + Attributes)

request:

```

curl -X GET
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/tag?phi_namespace={phi_namespace} -H "Accept: application/json" -H "Cookie: sid={sid}"

```

response:

```

{
  "namespace":"0785926b-9a28-4fcc-b0d9-e57488647403",
  "study_uid":"1.2.124.113532.10.236.100.70.20090407.180234.3203602",
  "tags":[
    {
      "tag":"(0000,0000)",
      "value":"VALUE"
    },
    ...
  ],
  "size":151795276,
  "image_count":285,
  "attachment_count":0
}

```

Tags: Transfer Attributes (MediaStorageSOP and TransferSyntax) for the study

request:

```
curl -X GET
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/attribute/transfer?phi_names
pace={phi_namespace} -H "Accept: application/json" -H "Cookie: sid={sid}"
```

response:

```
{"result":[
  {
    "tag":"(0002,0010)",
    "value":[
      "1.2.840.10008.1.2"
    ]
  },
  {
    "tag":"(0002,0002)",
    "value":[
      "1.2.840.10008.5.1.4.1.1.1"
    ]
  }
]}
```

Attachments

POST binary attachment: POST a new binary attachment for a study

Request:

```
curl -XPOST http://localhost:8500/api/v3/storage/study/{namespace}/{studyUid}/attachment \
--form data=@file.pdf -H "Cookie: sid={sid}" --form uploaded_by "Uploader Name"
```

with the following content-types:

```
-H "Content-type: application/pdf"
-H "Content-type: application/octet-stream"
```

Response:

```
{
  "namespace":"0785926b-9a28-4fcc-b0d9-e57488647403",
  "study_uid":"1.2.124.113532.10.236.100.70.20090407.180234.3203602",
  "attachment_guid":"0785926b-9a28-4fcc-b0d9-e574886423421",
  "attachment_version":"AKadopdaskdAOPKD212330qwQW",
  "source_label":"Uploader Name"
}
```

GET specific attachment: GET a specific attachment, by version

Request:

```
curl -XGET
```

```
http://localhost:8500/api/v3/storage/study/{namespace}/{studyUid}/attachment/{attachmentId}/version/{version}
```

Response

```
file, with correct content-type
```

GET latest attachment: GET a latest attachment, by version

Request:

```
curl -XGET http://localhost:8500/api/v3/storage/study/{namespace}/{studyUid}/attachment/latest
```

Response

```
file, with correct content-type
```

Orders/PHI

HL7 Order: Post a new order

request:

```
curl -XPOST http://localhost:8500/api/v3/storage/order/{order_namespace} -d @order.json -H "Content-type: application/json" -H "Cookie: sid={sid}"
```

where order.json HL7Message type:

```
{
  "message_id": "112",
  "patient": {
    "patient_name": "hello world",
    "patient_gender": "F",
    "patient_dob": "YYYYMMDD",
    "order_id": "1212",
    "study_date": "YYYYMMDD"
  },
  "hl7_raw_message": "qwqwqwqw"
}
```

response:

```
GUID
```

HL7 Order: Search orders for a Namespace

request:

```
curl -XGET http://localhost:8500/api/v3/storage/order/{order_namespace}?[patient_name|patient_gender|patient_dob|order_id|study_date]=value -H "Content-type: application/json" -H "Cookie: sid={sid}"
```

response:

```
{
  "result": [
    {
      "namespace": "1111",
      "attachment_guid": "50e2f19a-2c18-460f-9507-9d9544bef573",
      "attachment_version": "bYfSXdb0rPnzjfWsicYS2b0lnj4",
      "order_data": {
        "patient_name": "hello",
        "patient_gender": "F",
        "patient_dob": "YYYYMMDD",
        "order_id": "1212",
        "study_date": "YYYYMMDD"
      },
      "normalized_score": 0.2518424093723297
    }
  ]
}
```

PHI: Associate order to Study

request:

```
curl -XPUT http://localhost:8500/api/v3/storage/study/{ns}/{study_uid}/order/{order_namespace} -d
@order.json -H "Content-type: application/json" -H "Cookie: sid={sid}"
```

options:

you can disable notification by adding header "X-Dicomgrid-No-Notification":

```
curl -XPUT http://localhost:8500/api/v3/storage/study/{ns}/{study_uid}/order/{order_namespace} -d
@order.json -H "Content-type: application/json" -H "Cookie: sid={sid}" -H
"X-Dicomgrid-No-Notification: 1"
```

where order.json is RelevantData type:

```
{
  "patient_name": "hello world", // PID.5 PatientName(0010,0010)
  "patient_gender": "F", // PID.8 PatientSex(0010,0040)
  "patient_dob": "YYYYMMDD", // PID.7 PatientBirthDate(0010,0030)
  "patient_physical_address": "some address", // PID.11 DICOM(0010,1040)
  "order_id": "1212", // AccessionNumber(0008,0050)
  "order_description": "some description", // StudyDescription(0008,1030)
  "study_date": "YYYYMMDD", // StudyDate(0008,0020)
  "patient_mrn": "1121212" // PatientId (0010,0020)
  "referring_physician": "some physician" // ReferringPhysicianName (0008,0090)
  "extra_tags": "{\\"(0012,0001)\":\\"value\\"}" // JSON-formatted extra DICOM tags
}
```

response:

GUID (association ID)

GET Image-level Data

Thumbnail (JPEG): Gets jpeg of the thumbnail image (512x512, 75%)

request:

```
curl -X GET
```

```
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/image/{image_uid}/version/{version_id}/frame/{frame_id}/thumbnail -H "Accept: image/jpeg" -H "Cookie: sid={sid}" > thumb.jpeg
```

response:

JPEG file

Image (JPEG, FULL RES): Gets jpeg of full res image (original HxW, 95%)

request:

```
curl -X GET
```

```
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/image/{image_uid}/version/{version_id}/frame/{frame_id} -H "Accept: image/jpeg" -H "Cookie: sid={sid}" > full_res.jpeg
```

response:

JPEG file

Image (DICOM, DIAGNOSTIC): Original DICOM quality

request:

```
curl -X GET
```

```
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/image/{image_uid}/version/{version_id} -H "Accept: application/dicom" -H "Cookie: sid={sid}" > full_dicom.dcm
```

response:

DICOM file

Image (DICOM, DIAGNOSTIC): Original DICOM quality, with different namespace PHI

request:

```
curl -X GET
```

```
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/image/{image_uid}/version/{version_id}?phi_namespace={phi_namespace} -H "Accept: application/dicom" -H "Cookie: sid={sid}" > full_dicom.dcm
```

response:

DICOM file

Image PHI: Image PHI with original PHI namespace

request:

```
curl -XGET
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/image{image_uid}/version/{version_id}/phi -H "Accept: application/json" -H "Cookie: sid={sid}"
```

response:

```
{
  "tag": "1.3.6.1.4.1.25403.207458815553375.4024.20110824010956.67",
  "value": "1.3.6.1.4.1.25403.207458815553375.4024.20110824010956.67",
  "attr": [
    {
      "tag": "(0008,0050)",
      "value": "tt100000001"
    },
    {
      "tag": "(0008,1030)",
      "value": "ANatoly"
    },
    {
      "tag": "(0010,0010)",
      "value": "tolytest1"
    },
    {
      "tag": "(0010,0020)",
      "value": "tolytest1 12345678"
    },
    {
      "tag": "(0010,0030)",
      "value": "19320618"
    },
    {
      "tag": "(0010,0040)",
      "value": "M"
    }
  ]
}
```

Image PHI: Image PHI with a different PHI namespace

request:

```
curl -XGET
http://localhost:8500/api/v3/storage/study/{namespace}/{study_uid}/image{image_uid}/version/{version_id}/phi?phi_namespace={phi_namespace} -H "Accept: application/json" -H "Cookie: sid={sid}"
```

response:

```
{
  "tag": "1.3.6.1.4.1.25403.207458815553375.4024.20110824010956.67",
  "value": "1.3.6.1.4.1.25403.207458815553375.4024.20110824010956.67",
  "attr": [
    {
      "tag": "(0008,0050)",
      "value": "tt100000001"
    },
    {
      "tag": "(0008,1030)",
      "value": "ANatoly"
    },
    {
      "tag": "(0010,0010)",
      "value": "tolytest1"
    },
    {
      "tag": "(0010,0020)",
      "value": "tolytest1 12345678"
    },
    {
      "tag": "(0010,0030)",
      "value": "19320618"
    },
    {
      "tag": "(0010,0040)",
      "value": "M"
    }
  ]
}
```

POST/PUT Image Level Data

Image: Upload image to Storage

request:

```
curl -v http://localhost:8500/api/v3/storage/namespace/{ns}/image -XPOST --data-binary @file.dcm
-H "Content-Type: application/dicom" -H "Cookie: sid={sid}"
```

options:

along w/ payload, you can also pass source and destination AE Titles:

```
curl -v http://localhost:8500/api/v3/storage/namespace/{ns}/image -XPOST --data-binary @file.dcm
-H "Content-Type: application/dicom" -H "Cookie: sid={sid}" \
-H "X-Dicomgrid-Routing-Source-AETitle: AETITLESOURCE" -H
"X-Dicomgrid-Routing-Destination-AETitle: AETITLEDEST"
```

response:

```

{
  "namespace":"1111",
  "study_uid":"1.3.6.1.4.1.25403.207458815553375.4024.20110824010947.1",
  "image_uid":"1.3.6.1.4.1.25403.207458815553375.4024.20110824010948.185",
  "attr":[
    {
      "tag":"(0008,0018)",
      "value":"1.3.6.1.4.1.25403.207458815553375.4024.20110824010948.185"
    },
    {
      "tag":"(0008,0050)",
      "value":"tt100000001"
    },
    {
      "tag":"(0008,1030)",
      "value":"ANatoly"
    },
    {
      "tag":"(0008,103E)",
      "value":".625 ABD/PEL"
    },
    {
      "tag":"(0010,0010)",
      "value":"tolytest1"
    },
    {
      "tag":"(0010,0020)",
      "value":"tolytest112345678"
    },
    {
      "tag":"(0010,0030)",
      "value":"19320618"
    },
    {
      "tag":"(0010,0040)",
      "value":"M"
    },
    {
      "tag":"(0020,000D)",
      "value":"1.3.6.1.4.1.25403.207458815553375.4024.20110824010947.1"
    },
    {
      "tag":"(0020,000E)",
      "value":"1.3.6.1.4.1.25403.207458815553375.4024.20110824010948.171"
    }
  ],
  "image_version":"KMp0upxmpbbauOlbf6DqjisBiCo"
}

```

DELETE Study

Study: Delete a study and all related PHI, attachments, orders, images

Request:

```
curl -v -XDELETE http://localhost:8500/api/v3/storage/study/{namespace}/{studyUid} -H "Cookie: sid={sid}"
```

Response: HTTP 202/Accepted if removal was successful

Sample Code

Get Study Schema, C#

```
public string GetSchema(string studyNamespace, string phiNamespace, string studyUid)
{
    log.DebugFormat("Gathering schema for study {0}/{1} via phi_namespace {2}",
        studyNamespace, studyUid, phiNamespace);
    var requestUri = Configurations["storageUriRoot"];
    requestUri += string.Format("/study/{0}/{1}/schema?phi_namespace={2}",
        studyNamespace, studyUid, phiNamespace);
    HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(requestUri);
    request.Headers.Add("Authorization", "Basic " + _authorization);
    var responseString = string.Empty;
    using (var response = request.GetResponse())
    {
        responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();
    }
    return responseString;
}
```

Details

How are settings generated

Configuration for V3 Storage lives in 4 places:

1. Jetty config lives in GridV2WebServices/src/main/webapp/WEB-INF
2. Web config can be found in GridV2WebServices/src/resources/com/dicomgrid/webService
3. Task config lives in GridV2WebServices/src/test/templates
4. General config lives in GridV2WebServices/src/test/templates