Compass
DICOMweb Conformance Statement

LAUREL BRIDGE

Orchestrating Medical Imaging Workflow

Laurel Bridge Software, Inc.
302-453-0222
www.laurelbridge.com

Document Version: 3.5.1

Last Saved: 5/24/2023 4:47:00 PM

Table of Contents

1 Overview

Compass supports a subset of the RESTful web services specified in the DICOM PS3.18 - Web Services standard (commonly referred to as DICOMweb). Specifically, it supports the Store Transaction of the Studies Service and Resources (commonly referred to as STOW-RS, which stands for Store Over the Web – Restful Service). It also supports the Search Transaction of the Studies Service and Resources (commonly referred to as QIDO-RS, which stands for Query based on ID for DICOM Objects – Restful Service). It does not support the Retrieve Transaction, URI Service, Worklist Service, Non-Patient Instance Service, or any of the Capabilities Transactions.

2 Store Transaction (STOW-RS)

The Store Transaction (STOW-RS) is a RESTful web service for storing DICOM imaging data from a client to a server. Compass supports the Store Transaction both as a User Agent (web client) and as an Origin Server (web server).

2.1 Compass STOW-RS User Agent

The Store Transaction supports sending either DICOM Part 10 binary files directly or it supports the splitting of the DICOM information into metadata (represented in JSON or XML) and bulk data and sending these separately. The Compass STOW-RS User Agent supports sending all three variants, which are described in the following sections.

2.1.1 DICOM Part 10 File Store Requests

The Compass STOW-RS User Agent supports the following Content-Types for sending DICOM Part 10 binary files:

- multipart/related; type=application/dicom
- application/dicom

By default, Compass will send DICOM instances as a multipart HTTP request (containing one or more instances, each with content type multipart/related; type=application/dicom). Note that Compass can also optionally be configured to send each DICOM instance as a single part HTTP request (containing a single instance, with the content type application/dicom) by setting the request batch size equal to 1.

Any SOP class from the Compass DICOM Conformance Statement, Table 4.2.1.1-1, can be sent via this store transaction, using any of the transfer syntaxes from Table 4.2.1.1.1-1. Note that if transcoding needs to be performed (i.e., the instances will be sent out with a different transfer syntax encoding than they were received/stored as), the list of transfer syntaxes which cannot be transcoded applies.

2.1.2 JSON Metadata with Bulk Data Store Requests

When storing instances using JSON metadata and bulk data, a multipart HTTP request must be used. The following Content-Type is used for the request:

- multipart/related; type=application/dicom+json; transfer-syntax=*{transferSyntax}*

where *{transferSyntax}* is the transfer syntax used to encode any binary data as InlineBinary. Compass supports the following transfer syntax(es) for InlineBinary data:

- 1.2.840.10008.1.2.1 – Explicit VR Little Endian

The first multipart section (the DICOM metadata) contains a JSON array of DICOM instance(s). The JSON metadata itself is encoded in Unicode UTF-8. Compass supports the following Content-Type(s) for the JSON metadata section:

- application/dicom+json

The remaining multipart section(s) encode the pixel data element(s) for the instances described in the JSON metadata. Compass supports the following Content-Type(s) for the bulk data section(s):

- application/octet-stream
- image/jpeg

Within the JSON metadata, the pixel data element (7FE0, 0010) for each DICOM instance is encoded as either a Value or a BulkData property whose value is the BulkDataURI (relative URI) for that instance. This value will match the value of the Content-Location header for the matching bulk data section. (In practice, the bulk data sections will always be sent in the same order as the order of the instances in the JSON metadata.) The BulkDataURIs will always be unique within a Store Transaction request.

Compressed JPEG images received by Compass are assumed to be encoded as JPEG Baseline (i.e., Transfer Syntax = 1.2.840.10008.1.2.4.50) images. Compressed JPEG images sent by Compass can be sent using any of the transfer syntaxes from the Compass DICOM Conformance Statement, Table 4.2.1.1.1-1. Note that if transcoding needs to be performed (i.e., the instances will be sent out with a different transfer syntax encoding than they were received/stored as), the list of transfer syntaxes which cannot be transcoded applies.

Currently, Compass requires that the JSON metadata contains the Group 0028 (Image Pixel) attributes, even for compressed JPEG images.

2.1.3 XML Metadata with Bulk Data Store Requests

When storing instances using XML metadata and bulk data, a multipart HTTP request must be used. The following Content-Type is used for the request:

- multipart/related; type=application/dicom+xml; transfer-syntax=*{transferSyntax}*

where *{transferSyntax}* is the transfer syntax used to encode any binary data as InlineBinary. Compass supports the following transfer syntax(es) for InlineBinary data:

- 1.2.840.10008.1.2.1 – Explicit VR Little Endian

The first multipart section(s) (the DICOM metadata) contain(s) one or more XML objects in Native DICOM Model format, encoded one per section. The XML metadata itself is encoded in Unicode UTF-8. Compass supports the following Content-Type(s) for the XML metadata section(s):

- application/dicom+xml

The remaining multipart section(s) encode the pixel data element(s) for the instances described in the XML metadata. Compass supports the following Content-Type(s) for the bulk data section(s):

- application/octet-stream
- image/jpeg

Within the XML metadata, the pixel data element (7FE0, 0010) for each DICOM instance is encoded as either a Value or a BulkData element whose value is the BulkDataURI (relative URI) for that instance. This value will match the value of the Content-Location header for the matching bulk data section. (In practice, the bulk data sections will always be sent in the same order as the order of the instances in the XML metadata.) The BulkDataURIs will always be unique within a Store Transaction request

Compressed JPEG images sent by Compass can be sent using any of the transfer syntaxes from the Compass DICOM Conformance Statement, Table 4.2.1.1.1-1. Note that if transcoding needs to be performed (i.e., the instances will be sent out with a different transfer syntax encoding than they were received/stored as), the list of transfer syntaxes which cannot be transcoded applies.

Currently, Compass requires that the XML metadata contains the Group 0028 (Image Pixel) attributes, even for compressed JPEG images.

2.1.4 Store Response Handling

Compass supports the Store Instances Response Module (in the HTTP response body) in either JSON or XML format. It can send an Accept header with any of the following media types:

- application/json (default)
- application/dicom+json
- application/xml
- application/dicom+xml

Note that when sending the application/json and application/dicom+json Accept media types, Compass will expect the same response, a Store Instances Response Module (as defined in DICOM PS 3.18, Appendix I) encoded in JSON format (as defined in DICOM PS 3.18, Appendix F). Likewise, when sending the application/xml and application/dicom+xml Accept

media types, Compass will expect the same Store Instances Response Module encoded in XML format (as defined in DICOM PS 3.19, Appendix A.1).

On success (all instances stored successfully), Compass will expect an HTTP status code of 200 (= OK) and a Store Instances Response Module containing a list of the instances successfully stored (in the ReferencedSOPSequence).

On partial success (one or more instances stored successfully, one or more instances failed to store), Compass will expect an HTTP status code of 202 (= Accepted) and a Store Instances Response Module containing a list of the instances successfully stored (in the ReferencedSOPSequence) and a list of the instances that failed to store (in the FailedSOPSequence).

On failure (no instances were stored successfully), Compass will expect any other valid HTTP status code. If a Store Instances Response Module is present, it will be examined for a list of the instances that failed to store (in the FailedSOPSequence) and a list of DICOM failure codes not tied to a specific instance (in the OtherFailuresSequence).

2.2   Compass STOW-RS Origin Server

The Compass STOW-RS Origin Server functions as a proxy implementation which routes the incoming requests into the standard Compass DIMSE routing engine.

The Store Transaction supports sending either DICOM Part 10 binary files directly or it supports the splitting of the DICOM information into metadata (represented in JSON or XML) and bulk data and sending these separately. The Compass STOW-RS Origin Server supports accepting all three variants, which are described in the following sections.

2.2.1   DICOM Part 10 File Store Requests

The Compass STOW-RS Origin Server accepts Store Transactions with the following Content-Types for DICOM Part 10 binary files:

- multipart/related; type=application/dicom
- application/dicom

Compass can accept DICOM instances as multipart HTTP requests (containing one or more instances, each with content type multipart/related; type=application/dicom) or as single part HTTP requests (containing a single instance, with the content type application/dicom).

Any SOP class from the Compass DICOM Conformance Statement, Table 4.2.1.1-1, can be accepted via the Store Transaction, using any of the transfer syntaxes from Table 4.2.1.1.1-1. Note that if transcoding needs to be performed (i.e., if filtering has been configured), the list of transfer syntaxes which cannot be transcoded applies.

2.2.2 JSON Metadata with Bulk Data Store Requests

The Compass STOW-RS Origin Server accepts Store Transactions with the following Content-Type for JSON metadata plus bulk data multipart HTTP requests:

- multipart/related; type=application/dicom+json; transfer-syntax=*{transferSyntax}*

where *{transferSyntax}* is the transfer syntax used to encode any binary data as InlineBinary. Compass supports the following transfer syntax(es) for InlineBinary data:

- 1.2.840.10008.1.2.1 – Explicit VR Little Endian

The first multipart section (the DICOM metadata) must contain a JSON array of DICOM instance(s). The JSON metadata itself must be encoded in Unicode UTF-8. Compass supports the following Content-Type(s) for the JSON metadata section:

- application/dicom+json

The remaining multipart section(s) encode the bulk data for the instances described in the JSON metadata. Compass supports the following Content-Type(s) for the bulk data sections:

- application/octet-stream
- image/jpeg

Within the JSON metadata, the pixel data element (7FE0, 0010) for each DICOM instance must be encoded as either a Value or a BulkData property whose value is the BulkDataURI (relative URI) for that instance. This value must match the value of the Content-Location header for the matching bulk data section. The BulkDataURIs must be unique within a Store Transaction request.

Compressed JPEG images received by Compass are assumed to be encoded as JPEG Baseline (i.e., Transfer Syntax = 1.2.840.10008.1.2.4.50) images. Compressed JPEG images can be accepted in any of the transfer syntaxes from the Compass DICOM Conformance Statement, Table 4.2.1.1.1-1. Note that if transcoding needs to be performed (i.e., if filtering is configured), the list of transfer syntaxes which cannot be transcoded applies.

2.2.3 XML Metadata with Bulk Data Store Requests

The Compass STOW-RS Origin Server accepts Store Transactions with the following Content-Type for XML metadata plus bulk data multipart HTTP requests:

- multipart/related; type=application/dicom+xml; transfer-syntax=*{transferSyntax}*

where *{transferSyntax}* is the transfer syntax used to encode any binary data as InlineBinary. Compass supports the following transfer syntax(es) for InlineBinary data:

- 1.2.840.10008.1.2.1 – Explicit VR Little Endian

The first multipart section (the DICOM metadata) must contain an XML object in Native DICOM Model format. The XML metadata itself is encoded in Unicode UTF-8. Compass supports the following Content-Type(s) for the XML metadata section(s):

- application/dicom+xml

The remaining multipart section(s) can be either more XML metadata sections or bulk data sections encoding the bulk data for the instances described in earlier XML metadata sections, with the following caveats: 1) An XML metadata section must always precede its matching pixel data section, and 2) There must a one-to-one correspondence between XML metadata sections and bulk data sections. Compass supports the following Content-Type(s) for the bulk data sections:

- application/octet-stream
- image/jpeg

Within the XML metadata, the pixel data element (7FE0, 0010) for each DICOM instance is encoded as either a Value or a BulkData element whose value is the BulkDataURI (relative URI) for that instance. This value will match the value of the Content-Location header for the matching bulk data section. (In practice, the bulk data sections will always be sent in the same order as the order of the instances in the XML metadata.) The BulkDataURIs will always be unique within a Store Transaction request

Compressed JPEG images received by Compass are assumed to be encoded as JPEG Baseline (i.e., Transfer Syntax = 1.2.840.10008.1.2.4.50) images. Compressed JPEG images can be accepted in any of the transfer syntaxes from the Compass DICOM Conformance Statement, Table 4.2.1.1.1-1. Note that if transcoding needs to be performed (i.e., if filtering is configured), the list of transfer syntaxes which cannot be transcoded applies.

2.2.4 Store Responses

Compass supports the Store Instances Response Module (in the HTTP response body) in either JSON or XML format. It can accept an Accept header with any of the following media types:

- application/json (default)
- application/dicom+json
- application/xml
- application/dicom+xml

Note that for the application/json and application/dicom+json Accept media types, Compass will send the same response, a Store Instances Response Module (as defined in DICOM PS 3.18, Appendix I) encoded in JSON format (as defined in DICOM PS 3.18, Appendix F). Likewise, for the application/xml and application/dicom+xml Accept media types, Compass will send the same Store Instances Response Module encoded in XML format (as defined in DICOM PS 3.19, Appendix A.1).

On success (all instances stored successfully), Compass will send an HTTP status code of 200 (= OK) and a Store Instances Response Module containing a list of the instances successfully stored (in the ReferencedSOPSequence).

On partial success (one or more instances stored successfully, one or more instances failed to store), Compass will send an HTTP status code of 202 (= Accepted) and a Store Instances Response Module containing a list of the instances successfully stored (in the ReferencedSOPSequence) and a list of the instances that failed to store (in the FailedSOPSequence).

On failure, Compass will send one of the following HTTP status codes:

- 400 – Bad Request (error parsing request)
- 404 – Not Found (STOW-RS is not enabled)
- 409 – Conflict (all instances failed to store)
- 415 – Unsupported Media Type
- 500 – Internal Server Error (generic server error – see Compass logs for details)
- 503 – Service Unavailable (transient server out-of-resources error)

If a Store Instances Response Module is sent, it will contain a list of the instances that failed to store (in the FailedSOPSequence) and a list of the DICOM failure codes not tied to a specific instance (in the OtherFailuresSequence).

3   Query Transaction (QIDO-RS)

The Query Transaction (QIDO-RS) is a RESTful web service for querying stored DICOM instances on a server. Compass supports the Query Transaction both as a User Agent (web client) and as an Origin Server (web server).

3.1   Compass QIDO-RS User Agent

3.1.1   Query Requests

The Compass QIDO-RS User Agent supports querying of the following resources:

Resource	URI Template	Query Type
All Studies	/studies{?*search**}	Hierarchical
A Study's Series	/studies/{*study*}/series{?*search**}	Hierarchical
A Study's Instances	/studies/{*study*}/instances{?*search**}	Relational
All Series	/series{?*search**}	Relational

Resource	URI Template	Query Type
A Series' Instances	/studies/{*study*}/series/{*series*}/instances{*?search**}	Hierarchical
All Instances	/instances{*?search**}	Relational

Hierarchical queries should be supported by any origin server. Relational queries may not be supported by a particular origin server. Italicized terms have the following meanings:

- {*study*} indicates the StudyInstanceUID for a particular study
- {*series*} indicates the SeriesInstanceUID for a particular series
- {*?search**} indicates one or more query parameters can follow the '?', separated by an '&', as described below

The Compass QIDO-RS User Agent supports the following query parameters:

Term	Value
search	= *match* / *includefield*

match	= *8hexdigits* "=" *value* /
	    keyword "=" *value*
	 where:
	    8hexdigits is a DICOM attribute tag as 8 uppercase hex digits
	    keyword is an attribute keyword from DICOM PS 3.6, Table 6-1
	    value is the value to search for (can be wildcarded with '*')

includefield	= "includefield" "=" *8hexdigits* / *keyword* / "all"

The *fuzzymatching*, *limit*, and *offset* query parameters are not supported. UID list matching is not currently supported. Date range and time range matching are supported for attributes with DICOM VRs of DA, TM, and DT.

The Query Transaction does not support querying at a patient level. This means that patient-level attributes are considered study-level attributes for the purpose of querying. The Compass QIDO-RS User Agent supports the following query attributes:

Level	Attribute Name (Keyword)	Attribute Tag
Study	StudyDate	00080020

Level	Attribute Name (Keyword)	Attribute Tag
	StudyTime	00080030
	AccessionNumber	00080050
	ModalitiesInStudy	00080061
	SOPClassesInStudy	00080062
	ReferringPhysicianName	00080090
	StudyDescription	00081030
	NameOfPhysiciansReadingStudy	00081060
	AdmittingDiagnosesDescription	00081080
	PatientName	00100010
	PatientID	00100020
	IssuerOfPatientID	00100021
	PatientBirthDate	00100030
	PatientSex	00100040
	OtherPatientNames	00101001
	PatientAge	00101010
	PatientSize	00101020
	PatientWeight	00101030
	EthnicGroup	00102160
	Occupation	00102180
	AdditionalPatientHistory	001021B0
	PatientComments	00104000
	StudyInstanceUID	0020000D
	StudyID	00200010
Series	Modality	00080060
	SeriesInstanceUID	0020000E
	SeriesNumber	00200011
	PerformedProcedureStepStartDate	00400244
	PerformedProcedureStepStartTime	00400245
Instance	SOPClassUID	00080016
	SOPInstanceUID	00080018

Level	Attribute Name (Keyword)	Attribute Tag
	InstanceNumber	00200013

3.1.2 Query Response Handling

The Compass QIDO-RS User Agent supports Query Transaction responses in either JSON or XML format. It can send an Accept header with any of the following media types:

- application/json (default)
- application/dicom+json
- application/xml
- application/dicom+xml

Note that when sending the application/json and application/dicom+json Accept media types, Compass will expect responses encoded in JSON format (as defined in DICOM PS 3.18, Appendix F). Likewise, when sending the application/xml and application/dicom+xml Accept media types, Compass will expect responses encoded in XML format (as defined in DICOM PS 3.19, Appendix A.1).

On success, Compass will expect an HTTP status code of either 200 (= OK, meaning the query was successful, and results were returned) or 204 (= No Content, meaning the query was successful, but no results were returned). For status code 200, the response will be parsed for the encoded instances. For status code 204, the response (if any) will not be parsed.

On failure, Compass will expect any other valid HTTP status code. Any payload describing the error will be logged.

3.2 Compass QIDO-RS Origin Server

The Compass QIDO-RS Origin Server functions as a proxy implementation which routes the incoming requests into the standard Compass DIMSE routing engine.

3.2.1 Query Requests

The Compass QIDO-RS Origin Server supports querying of the following resources:

Resource	URI Template	Query Type
All Studies	/studies{?*search**}	Hierarchical
A Study's Series	/studies/{*study*}/series{?*search**}	Hierarchical
A Study's Instances	/studies/{*study*}/instances{?*search**}	Relational

Resource	URI Template	Query Type
All Series	/series{?*search**}	Relational
A Series' Instances	/studies/{*study*}/series/{*series*}/instances{?*search**}	Hierarchical
All Instances	/instances{?*search**}	Relational

Note that both the Hierarchical and Relational query types are supported. Italicized terms have the following meanings:

- {*study*} indicates the StudyInstanceUID for a particular study
- {*series*} indicates the SeriesInstanceUID for a particular series
- {?*search**} indicates one or more query parameters can follow the '?', separated by an '&', as described below

The Compass QIDO-RS Origin Server supports the following query parameters:

Term	Value
search	= *match* / *includefield*

match	= *8hexdigits* "=" *value* /
	    keyword "=" *value*
	 where:
	    8hexdigits is a DICOM attribute tag as 8 uppercase hex digits
	    keyword is an attribute keyword from DICOM PS 3.6, Table 6-1
	    value is the value to search for (can be wildcarded with '*')

includefield	= "includefield" "=" *8hexdigits* / *keyword* / "all"

The *fuzzymatching*, *limit*, and *offset* query parameters are not supported. UID list matching is not currently supported. Date range and time range matching are supported for attributes with DICOM VRs of DA, TM, and DT.

The Query Transaction does not support querying at a patient level. This means that patient-level attributes are considered study-level attributes for the purpose of querying. The Compass QIDO-RS Origin Server supports the following query attributes:

Level	Attribute Name (Keyword)	Attribute Tag
Study	StudyDate	00080020
	StudyTime	00080030
	AccessionNumber	00080050
	ModalitiesInStudy	00080061
	SOPClassesInStudy	00080062
	ReferringPhysicianName	00080090
	StudyDescription	00081030
	NameOfPhysiciansReadingStudy	00081060
	AdmittingDiagnosesDescription	00081080
	PatientName	00100010
	PatientID	00100020
	IssuerOfPatientID	00100021
	PatientBirthDate	00100030
	PatientSex	00100040
	OtherPatientNames	00101001
	PatientAge	00101010
	PatientSize	00101020
	PatientWeight	00101030
	EthnicGroup	00102160
	Occupation	00102180
	AdditionalPatientHistory	001021B0
	PatientComments	00104000
	StudyInstanceUID	0020000D
	StudyID	00200010
Series	Modality	00080060
	SeriesInstanceUID	0020000E
	SeriesNumber	00200011
	PerformedProcedureStepStartDate	00400244
	PerformedProcedureStepStartTime	00400245
Instance	SOPClassUID	00080016

Level	Attribute Name (Keyword)	Attribute Tag
	SOPInstanceUID	00080018
	InstanceNumber	00200013

3.2.2 Query Responses

The Compass QIDO-RS Origin Server supports Query Transaction responses in either JSON or XML format. It can accept an Accept header with any of the following media types:

- application/json (default)
- application/dicom+json
- application/xml
- application/dicom+xml

Note that for the application/json and application/dicom+json Accept media types, Compass will send the same responses encoded in JSON format (as defined in DICOM PS 3.18, Appendix F). Likewise, for the application/xml and application/dicom+xml Accept media types, Compass will send the same responses encoded in XML format (as defined in DICOM PS 3.19, Appendix A.1).

On success, Compass will send an HTTP status code of either 200 (= OK, meaning the query was successful, and results were returned) or 204 (= No Content, meaning the query was successful, but no results were returned). For status code 200, the response will contain the encoded responses. For status code 204, the response will not contain any encoded responses.

When acting in full proxy mode, Compass will forward as many responses as are returned by the destination origin server. When acting in Compass Cache proxy mode, the maximum number of responses is configurable.

On failure, Compass will send one of the following HTTP status codes:

- 400 – Bad Request (error parsing request)
- 404 – Not Found (QIDO-RS is not enabled)
- 415 – Unsupported Media Type
- 500 – Internal Server Error (generic server error – see Compass logs for details)
- 503 – Service Unavailable (transient server out-of-resources error)

4 Retrieve Transaction (WADO-RS)

The Retrieve Transaction (WADO-RS) is a RESTful web service for retrieving stored DICOM instances on a server. Compass supports the Retrieve Transaction both as a User Agent (web client) and as an Origin Server (web server).

4.1 Compass WADO-RS User Agent

4.1.1 Retrieve Requests

The Compass WADO-RS User Agent supports querying of the following DICOM resources:

Resource	URI Template
Study	/studies/{*study*}
Series	/studies/{*study*}/series/{*series*}
Instance	/studies/{*study*}/series/{*series*}/instances/{*instance*}

Italicized terms have the following meanings:

- {*study*} indicates the StudyInstanceUID for a particular study
- {*series*} indicates the SeriesInstanceUID for a particular series
- {*instance*} indicates the SOPInstanceUID for a particular instance

Metadata resources, Rendered resources, Thumbnail resources, and bulk data resources are not currently supported.

Query parameters are not currently supported.

The Retrieve Transaction does not support querying at a patient level.

4.1.2 Retrieve Response Handling

The Compass WADO-RS User Agent supports Retrieve Transaction responses in DICOM format only. It will send an Accept header with the following media type:

- application/dicom

The Compass WADO-RS User Agent also supports the Transfer Syntax parameter, which allows the client to request that the returned data set be returned encoded in a particular transfer syntax, as in one of the following:

- application/dicom; transfer-syntax=*{transferSyntax}*
- application/dicom; transfer-syntax=*

The valid values for the *{transferSyntax}* are:

- None – Meaning no transfer syntax parameter is sent
- Any – Meaning "*" is sent
- 1.2.840.10008.1.2 – Implicit VR Little Endian
- 1.2.840.10008.1.2.1 – Explicit VR Little Endian

- 1.2.840.10008.1.2.2 – Explicit VR Big Endian (Retired)
- 1.2.840.10008.1.2.4.70 – JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection Value 1])
- 1.2.840.10008.1.2.4.50 – JPEG Baseline (Process 1)
- 1.2.840.10008.1.2.4.51 – JPEG Extended (Process 2 & 4)
- 1.2.840.10008.1.2.4.57 – JPEG Lossless, Non-Hierarchical (Process 14)
- 1.2.840.10008.1.2.4.80 – JPEG-LS Lossless
- 1.2.840.10008.1.2.4.81 – JPEG-LS Lossy (Near-Lossless)
- 1.2.840.10008.1.2.4.90 – JPEG 2000 (Lossless Only)
- 1.2.840.10008.1.2.4.91 – JPEG 2000 (Lossy)
- 1.2.840.10008.1.2.4.92 – JPEG 2000 Part 2 Multi-component (Lossless Only)
- 1.2.840.10008.1.2.4.93 – JPEG 2000 Part 2 Multi-component
- 1.2.840.10008.1.2.5 – RLE Lossless

On success, Compass will expect an HTTP status code of 200 (= OK, meaning the query was successful and results were returned). It will then parse the response data sets.

On failure, Compass will expect any other valid HTTP status code. Any payload describing the error will be logged.

4.2 Compass WADO-RS Origin Server

The Compass WADO-RS Origin Server functions as a proxy implementation which routes the incoming requests into the standard Compass DIMSE routing engine.

4.2.1 Retrieve Requests

The Compass WADO-RS Origin Server supports querying of the following DICOM resources:

Resource	URI Template
Study	/studies/{*study*}
Series	/studies/{*study*}/series/{*series*}
Instance	/studies/{*study*}/series/{*series*}/instances/{*instance*}

The Compass WADO-RS Origin Server supports querying of the following Metadata resources:

Resource	URI Template
Study	/studies/{*study*}/metadata

Resource	URI Template
Series	/studies/{*study*}/series/{*series*}/metadata
Instance	/studies/{*study*}/series/{*series*}/instances/{*instance*}/metadata

The Compass WADO-RS Origin Server supports querying of the following bulk data resource:

Resource	URI Template
Instance	/studies/{*study*}/series/{*series*}/instances/{*instance*}/bulkdata

Italicized terms have the following meanings:

- {*study*} indicates the StudyInstanceUID for a particular study
- {*series*} indicates the SeriesInstanceUID for a particular series
- {*instance*} indicates the SOPInstanceUID for a particular instance

Rendered resources and Thumbnail resources are not currently supported.

Query parameters are not currently supported.

The Retrieve Transaction does not support querying at a patient level.

4.2.2 Retrieve Responses

The Compass WADO-RS Origin Server supports Retrieve Transaction responses in DICOM format. It will only accept Accept headers with the following media type for DICOM resources:

- application/dicom

The Compass WADO-RS Origin Server supports Retrieve Transaction responses in Metadata format. It will only accept Accept headers with the following media types for Metadata resources:

- application/dicom+json
- application/dicom+xml

The Compass WADO-RS Origin Server supports Retrieve Transaction responses in bulk data format. It will only accept Accept headers with the following media types for bulk data resources:

- image/jpeg
- application/octet-stream

The Compass WADO-RS Origin Server also supports the Transfer Syntax parameter of the Accept header, which allows the client to request that the returned data set be returned encoded in a particular transfer syntax, as in one of the following:

- application/dicom; transfer-syntax=*{transferSyntax}*
- application/dicom; transfer-syntax=*

The supported values for the *{transferSyntax}* are:

- None – Meaning no transfer syntax parameter was sent
- Any – Meaning "*" was sent
- 1.2.840.10008.1.2 – Implicit VR Little Endian
- 1.2.840.10008.1.2.1 – Explicit VR Little Endian
- 1.2.840.10008.1.2.2 – Explicit VR Big Endian (Retired)
- 1.2.840.10008.1.2.4.70 – JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection Value 1])
- 1.2.840.10008.1.2.4.50 – JPEG Baseline (Process 1)
- 1.2.840.10008.1.2.4.51 – JPEG Extended (Process 2 & 4)
- 1.2.840.10008.1.2.4.57 – JPEG Lossless, Non-Hierarchical (Process 14)
- 1.2.840.10008.1.2.4.80 – JPEG-LS Lossless
- 1.2.840.10008.1.2.4.81 – JPEG-LS Lossy (Near-Lossless)
- 1.2.840.10008.1.2.4.90 – JPEG 2000 (Lossless Only)
- 1.2.840.10008.1.2.4.91 – JPEG 2000 (Lossy)
- 1.2.840.10008.1.2.4.92 – JPEG 2000 Part 2 Multi-component (Lossless Only)
- 1.2.840.10008.1.2.4.93 – JPEG 2000 Part 2 Multi-component
- 1.2.840.10008.1.2.5 – RLE Lossless

On success, Compass will send an HTTP status code of 200 (= OK, meaning the query was successful, and results were returned). The response will contain the encoded DICOM Part 10 data sets.

The WADO-RS Origin Server allows the maximum number of responses to be configured via the source configuration advanced properties.

On failure, Compass will send one of the following HTTP status codes:

- 400 – Bad Request (error parsing request)
- 404 – Not Found (WADO-RS is not enabled)
- 406 – Not Acceptable (the requested media type is not supported)
- 415 – Unsupported Media Type
- 500 – Internal Server Error (generic server error – see Compass logs for details)
- 503 – Service Unavailable (transient server out-of-resources error)