Docstore URL Interface Specification

Introduction

Docstore provides a RESTful webservice api as a modern interface to query and retrieve data from the Docstore using standard HTTP and XML.

Most of the xml data is returned along with a url to an xsl stylesheet allowing the xml data to be viewed in a web browser. Although this is convenient, the real power of Docstore is unlocked when the XML data is consumed by other Web sites and services. This document describes the xml data available for consumption.

HTTP Methods

The HTTP protocol has 4 methods: GET, POST, PUT and DELETE. Most Web page requests use the GET method, submitting forms or uploading files use the POST method and the other two are rarely used. A RESTful webservice makes use of all four of these Methods, mapping them to distinct actions:

Method	Action
GET	Retrieve the resource
POST	Create a new resource
PUT	Replace/Update the resource
DELETE	Delete the resource

(For more information on REST, see:
http://en.wikipedia.org/wiki/Representational_state_transfer)

Although an excellent idea in theory, in practice it is often inconvenient or impossible to use any method except GET. To make life easier, Docstore offers a parameter that will force it to

treat a GET request as a POST: `treatGetAsPost=true`. Any GET request that has this value in the URL will be treated as a POST request.
E.g.
A HTTP GET request to the url:
https://SERVER:6443/docStore/store/Docstore%20Demo/document/35/key/Key1/?keyValue=keyValue1&treatGetAsPost=true
Will set the value of Key1 to "keyValue1" for document 35.

DocID replacement

A number of the Interface URLs require the Document ID to be used [DocID], however this can usually only be discovered using a separate HTTP request, and the parsing of the returned XML
E.g.
https://SERVER:6443/docStore/store/store1/document/q[bbbbbb]=cccccc&q[dddddd]=eeeeee
Will return XML containing <docid>35</docid>, which could then be used in a separate call to access the stored document:
https://SERVER:6443/docStore/store/Docstore%20Demo/document/35/

Docstore provides an alternative url option that resolves the Document ID from key-value pairs, so the url:

https://SERVER:6443/docStore/store/DOCSTORE1/document/q%5Bkey1%5D=value1&q%5BdocType%5D=docType1/key/KEYNAME/

would be resolved to the desired url:

https://SERVER:6443/docStore/store/DOCSTORE1/document/123/key/KEYNAME/

Note: Both the keys and values will need to be percent encoded BUT NOT the = separators between keys and values, nor the & separator between key value pairs.

Example:

String url = "https://SERVER:6443/docStore/store/DOCSTORE1/document/" +

```
encodeURIComponent("q[key1]") +
"=" + encodeURIComponent("value1") +
"&" + encodeURIComponent("q[docType]") +
"=" + encodeURIComponent("docType1") +
"/key/KEYNAME";
```

A sample encodeURIComponent function is listed in the Appendix

HTTPS

HTTPS is available in CHTTPD v4.04 and above which was released with CPPD 6.2.80
By default HTTPS uses the port 6443 and HTTP port 6400.

Authentication

There are several ways to provide credentials to Docstore:

The username and password can be provided using HTTP authentication.

This is the dialog box that pops up when a browser is used to log onto Docstore. It is also the values that can be passed to the open() method of a XMLHttpRequest object.

The username and password can be set in the cookies:

Note: The chttpd.* cookie values were not checked between v2.84 and v4.08

```
chttpd.user = USERNAME
chttpd.passwordHex = AABBCCDDEEFF0011
```

or

```
chttpd.user = USERNAME
chttpd.password = PASSWORD
```

or

CPPD = USERNAME|AABBCCDDEEFF0011 – this is System21 Workspace Single signon method

or

User = USERNAME
Pass = PASSWORD

The username and password can be passed in the URL

e.g.
`https://SERVER:6443/diagnostics/?user=USERNAME&passwordHex=AABBCCDDEEFF0011`

Note 1: The server value "Authentication Key" must be set with a 16 character hex value, eg: 0E329232EA6D0D73.
Note 2: The password must be hex encoded. The simplest way to get this is to log in as an administrator and visit:
https://SERVER:6443/diagnostics/?encodeString=PASSWORDTOENCODE
then look for the encodedString value
Note 3: in Javascript the credentials can be passed using:

```
xhr.setRequestHeader("Authorization", "Basic " + btoa(username + ":" + password))
```

Login URL

https://SERVER:6443/login can be used to log into the server with only a result code and human readable result.

Options - silent
The URL https://SERVER:6443/login/?silent=true does not return the human readable result.

Options - redir
https://SERVER:6443/login/?redir=http://redir.to/url
The browser session will be redirected to the url specified in the redir parameter.

Note that the url to redirect to should be processed with encodeURIComponent() or similar before being used. NOT EncodeURI or urlencode.

eg to redirect to:

https://SERVER:6443/files/folder/war/file.html?q[STATUS]=NEW&q[LCOD]=BOB

The correct url would be:
https://SERVER:6443/login/?redir=http%3A%2F%2F1server%3A6400%2Ffiles%2Fwar%2Ffile.html%3Fq%5BSTATUS%5D%3DNEW%26q%5BLCOD%5D%3DBOB

Return Values

Docstore uses the HTTP Response Status Code to indicate success or failure. A return value of 200 to 299 indicates success, any other value indicates a failure.
An error description is returned in the HTML that is returned in the HTTP Response body. The description can be retrieved using the xpath query:

```
selectSingleNode("//*[@id='statusDescription']")
```

Example Success Response

```
<!DOCTYPE html>
<html lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>200 OK</title>
  <style type="text/css">
    body{font-family:Verdana,sans-serif;font-size:9pt}
    h3{font-size:13pt}
    pre{font-family:Courier New,cour}
  </style>
</head>
<body>
      <br />
      <h3>200 OK</h3>
      <p id="statusDescription">Document Keys successfully updated from
      document 256</p>
      <div class='server_signature'><hr />chttpd (debug) at Cobwebb
      webservice</div>
</body>
</html>
```

Example Failure Response

```
<!DOCTYPE html>
<html lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>500 Internal Server Error</title>
  <style type="text/css">
    body{font-family:Verdana,sans-serif;font-size:9pt}
    h3{font-size:13pt}
    pre{font-family:Courier New,cour}
  </style>
</head>
<body>
      <br />
      <h3>500 Internal Server Error</h3>
      <p id="statusDescription">Failed to insert new keys</p>
      <div class='server_signature'><hr />chttpd (debug) at Cobwebb
      webservice</div>
</body>
</html>
```

Interface URLs

Docstore Homepage

https://SERVER:6443/docStore/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStore.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <stores url="http://SERVER:6400/docStore/store/" />
    <log url="http://SERVER:6400/docStore/log/" />
  </docStore>
</CPPD>
```

The Docstore homepage.

Docstore Log

https://SERVER:6443/docStore/log/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreLog.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <log>
      <entry>
        <id>312</id>
        <dattim>2010-10-12-15.49.15</dattim>
        <status>S</status>
        <statusDesc>Document stored. ID: 35</statusDesc>
        <path>/test/</path>
        <name>PO124572</name>
        <ext>PDF</ext>
        <docStore>DocStore Demo</docStore>
        <docType>Purchase Orders</docType>
        <docMimeType>1</docMimeType>
        <description>PO: 124572 for ABC SUPPLIES LTD(S000001) on
23/08/10</description>
        <keyValue1>124572</keyValue1>
        <keyValue2>S000001</keyValue2>
        <keyValue3>ABC SUPPLIES LTD</keyValue3>
        <keyValue4>23/08/10</keyValue4>
      </entry>
    </log>
  </docStore>
</CPPD>
```

Lists the most recent submissions to the Docstore along with the outcomes.

Docstore Document Store List

https://SERVER:6443/docStore/store/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreStores.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <library>DOCSTORE09</library>
      <description>DocStore Demonstration of CPPD / DocStore</description>
    </store>
    <store>…</store>
  </docStore>
</CPPD>
```

Lists all the document stores in this Docstore.

Creating a Docstore

https://SERVER:6443/config/docStore/store/

This is the URL used to submit a document to the Docstore. The http method must be POST unless `treatGetAsPost=true`

Parameters	Description
docStoreName	The name of the Docstore
docStoreLibrary	The IBM i Library for the Docstore (Max: 10 Chars)
docStoreDescription	a Description

The xhtml response would be like the sample output below:

```
<!DOCTYPE html>
<html lang="en" xml:lang="en">
<head>
      <meta charset="utf-8" />
      <title>200 OK</title>
</head>
<body>
      <br />
      <h3>200 OK</h3>
      <p id="statusDescription">Docstore TEST Created.</p>
      <div class='server_signature'>
      <hr />chttpd at Cobwebb webservice</div>
</body>
</html>
```

Individual Document Store Homepage

https://SERVER:6443/docStore/store/Docstore%20Demo/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreStore.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <library>DOCSTORE</library>
      <description>DocStore Demonstration of
CPPD/DocStore</description>
      <keys
url="http://SERVER:6400/docStore/store/DocStore%20Demo/key/" />
      <docTypes
url="http://SERVER:6400/docStore/store/DocStore%20Demo/docType/" />
      <docTypeKeys
url="http://SERVER:6400/docStore/store/DocStore%20Demo/docTypeKey/"
/>
      <documents
url="http://SERVER:6400/docStore/store/DocStore%20Demo/document/" />
    </store>
  </docStore>
</CPPD>
```

Lists the main attributes of this document Store. When viewed in a web browser, this is the url for the main search page for this document Store.

Document Store Keys

https://SERVER:6443/docStore/store/Docstore%20Demo/key/

```
<?xml version="1.0" encoding="utf-8"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <docStoreKeys>
        <docStoreKey>
          <id>4</id>
          <name>APINVNO</name>
          <description>Supplier Invoice No.</description>
          <type>CHAR</type>
          <size>10</size>
          <scale>0</scale>
        </docStoreKey>
        <docStoreKey>
          …
        </docStoreKey>
      </docStoreKeys>
    </store>
  </docStore>
</CPPD>
```

Lists all the Docstore Keys for this document Store.

Creating a Docstore Key

https://SERVER:6443/config/docStore/store/Docstore%20Demo/key/

This is the URL used to submit a document to the Docstore. The http method must be POST unless `treatGetAsPost=true`

Parameters	Description
keyName	The name of the Key (must begin with an alpha character)
keyDescription	a Description
keyType	The key type (See: key types)
keySize	The key size
keyScale	The key scale

The xhtml response would be like the sample output below:

```
<!DOCTYPE html>
<html lang="en" xml:lang="en">
<head>
      <meta charset="utf-8" />
      <title>200 OK</title>
      </head>
<body><br />
      <h3>200 OK</h3>
      <p id="statusDescription">Key: testKey created successfully</p>
      <div class='server_signature'>
      <hr />chttpd at Cobwebb webservice</div>
</body>
</html>
```

Document Store Document Types

https://SERVER:6443/docStore/store/Docstore%20Demo/docType/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl"
href="/files/docStoreDocTypes.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <documentTypes>
        <documentType>
          <id>3</id>
          <name>Purchase Orders</name>
          <description>Purchase Orders</description>
          <rootFolder>/cppd/DocStore/DocStore Demo/Purchase
          Orders</rootFolder>
          <autoName>TRUE</autoName>
          <lifeSpan>0</lifeSpan>
        </documentType>
      </documentTypes>
    </store>
  </docStore>
</CPPD>
```

Lists all the Document Types for this document Store.

Creating a Docstore Document Type

https://SERVER:6443/config/docStore/store/Docstore%20Demo/docType/

This is the URL used to create a Document Type. The http method must be POST unless `treatGetAsPost=true`

Parameters	Description
docTypeName	The name of the Document Type
docTypeDescription	a Description
defaultDocTypeRootFolder	TRUE/FALSE (Default: TRUE)
docTypeRootFolder	(Only required if defaultDocTypeRootFolder is FALSE)
docTypeAutoName	TRUE/FALSE (Default: FALSE)
hideUnauthorisedDocs	TRUE/FALSE (Default: FALSE)
docTypeDuplicateHandling	OVERWRITE/VERSION (Default: OVERWRITE)
docTypeLifeSpan	The lifespan of the document
docTypeDefaultAuthorisationList	The default authorisation list
docTypeProxyAuthObjectPath	The proxy auth object path

docTypeDocHandlingUsername	The document handling username

The xhtml response would be like the sample output below:

```
<!DOCTYPE html>
<html lang="en" xml:lang="en">
<head>
      <meta charset="utf-8" />
      <title>200 OK</title>
      </head>
<body><br />
      <h3>200 OK</h3>
      <p id="statusDescription">Document Type testDocType Created.</p>
      <div class='server signature'>
      <hr />chttpd at CoБwebb webservice</div>
</body>
</html>
```

Document Store DocType Keys

https://SERVER:6443/docStore/store/Docstore%20Demo/docTypeKey/

```
<?xml version="1.0" encoding="utf-8"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <documentTypeKeys>
        <documentTypeKey>
          <docTypeID>2</docTypeID>
          <keyTableID>4</keyTableID>
          <sortIndex>10</sortIndex>
          <keyRequired>1</keyRequired>
          <keyUnique>0</keyUnique>
        </documentTypeKey>
        <documentTypeKey>…</documentTypeKey>
      </documentTypeKeys>
      <documentTypes>
        <documentType>
          <id>3</id>
          <name>Purchase Orders</name>
          <description>Purchase Orders</description>
          <rootFolder>/cppd/DocStore/DocStore Demo/Purchase
          Orders</rootFolder>
          <autoName>TRUE</autoName>
          <lifeSpan>0</lifeSpan>
        </documentType>
        <documentType>…</documentType>
      </documentTypes>
      <docStoreKeys>
        <docStoreKey>
          <id>4</id>
          <name>APINVNO</name>
          <description>Supplier Invoice No.</description>
          <type>CHAR</type>
          <size>10</size>
          <scale>0</scale>
        </docStoreKey>
        <docStoreKey>…</docStoreKey>
      </docStoreKeys>
    </store>
  </docStore>
</CPPD>
```

Describes which Docstore Keys are assigned to which Document Types. Just for assistance it also duplicates the output of the previous two urls and lists all the keys and document types.

Document Store Document Type Homepage

https://SERVER:6443/docStore/store/Docstore%20Demo/docType/Purchase%20Orders/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreDocType.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <documentType>
        <id>3</id>
        <name>Purchase Orders</name>
        <description>Purchase Orders</description>
        <rootFolder>/cppd/DocStore/DocStore Demo/Purchase
        Orders</rootFolder>
        <autoName>TRUE</autoName>
        <lifeSpan>0</lifeSpan>
        <docTypeKeys
url="http://SERVER:6400/docStore/store/DocStore%20Demo/docType/Purchase%20Orders/docTypeKey/">
          <docTypeKey>
            <id>7</id>
            <name>PONo</name>
            <description>Purchase Order No.</description>
            <type>CHAR</type>
            <size>10</size>
            <scale>0</scale>
            <index>10</index>
            <required>TRUE</required>
            <unique>FALSE</unique>
          </docTypeKey>
          <docTypeKey>...</docTypeKey>
        </docTypeKeys>
      </documentType>
    </store>
  </docStore>
</CPPD>
```

Lists the main attributes of this Document Type. When viewed in a web browser, this is the url for the main search page for this Document Type.

Document Store Document Type Keys

https://SERVER:6443/docStore/store/Docstore%20Demo/docType/Purchase%20Orders/docTypeKey/

```
<?xml version="1.0" encoding="utf-8"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <documentType>
        <name>Purchase Orders</name>
        <docTypeKey>
          <id>7</id>
          <name>PONo</name>
          <description>Purchase Order No.</description>
          <type>CHAR</type>
          <size>10</size>
          <scale>0</scale>
          <index>10</index>
          <required>TRUE</required>
          <unique>FALSE</unique>
        </docTypeKey>
        <docTypeKey>...</docTypeKey>
      </documentType>
      <docStoreKeys>
        <docStoreKey>
          <id>4</id>
          <name>APINVNO</name>
          <description>Supplier Invoice No.</description>
          <type>CHAR</type>
          <size>10</size>
          <scale>0</scale>
        </docStoreKey>
        <docStoreKey>...</docStoreKey>
      </docStoreKeys>
    </store>
  </docStore>
</CPPD>
```

Describes the Docstore Keys that are assigned to this Document Type.

Associating a key with a DocType

https://SERVER:6443/config/docStore/store/Docstore%20Demo/docType/Purchase%20Orders/docTypeKey/

This is the URL used to associate a Docstore Key with a DocType. The http method must be POST unless `treatGetAsPost=true`

Parameters	Description
keyName	The name of the key to associate with the docType
keyRequired	TRUE/FALSE (Default: FALSE)
keyUnique	TRUE/FALSE (Default: FALSE)

The xml response would be like the sample output below:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreDocTypeKeysConfig.xsl"?>
<CPPD>
    <Library>CPPD</Library>
    <filenameURLWorkaround>TRUE</filenameURLWorkaround>
    <docStore>
        <store>
            <name>testds</name>
            <documentTypes>
                <documentType>
                    <id>1</id>
                    <name>testDocType</name>
                    <description>test description</description>
                    <rootFolder>/cppd/DocStore/testds/testDocType</rootFolder>
                    <autoName>TRUE</autoName>
                    <lifeSpan>100</lifeSpan>
                    <proxyAuthorityObjectPath></proxyAuthorityObjectPath>
                    <documentHandlingUsername></documentHandlingUsername>
                    <hideUnauthorisedDocs>TRUE</hideUnauthorisedDocs>
                    <defaultAuthorisationList></defaultAuthorisationList>
                    <duplicateHandling>OVERWRITE</duplicateHandling>
                </documentType>
            </documentTypes>
            <docStoreKeys>
                <docStoreKey>
                    <id>1</id>
                    <name>testKey</name>
                    <description>test key description</description>
                    <type>VARCHAR</type>
                    <size>32</size>
                    <scale>0</scale>
                </docStoreKey>
            </docStoreKeys>
            <documentTypeKeys></documentTypeKeys>
        </store>
    </docStore>
</CPPD>
```

Document Store Document List

https://SERVER:6443/docStore/store/Docstore%20Demo/document/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreDocuments.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <document
url='http://SERVER:6400/docStore/store/DocStore%20Demo/document/35/'
authorised='TRUE'>
        <docid>35</docid>
        <doctypeid>3</doctypeid>
        <docPath>/cppd/DocStore/DocStore Demo/Purchase
        Orders</docPath>
        <docFileName>DTLVTC4BPY.PDF</docFileName>
        <docOriginalFileName>PO124572.PDF</docOriginalFileName>
        <docMimeType>1</docMimeType>
        <docTimeAdded>2010-10-12-15.49.15</docTimeAdded>
        <docTimeAccessed>2010-10-28-15.36.31</docTimeAccessed>
        <docAccessCount>2</docAccessCount>
        <docDescription>PO: 124572 for ABC SUPPLIES LTD(S000001) on
        23/08/10</docDescription>
        <docOwner>DANIEL</docOwner>
      </document>
      <pagination>
        <start>1</start>
        <count>20</count>
        <total>3</total>
      </pagination>
      <filters></filters>
    </store>
  </docStore>
</CPPD>
```

Lists all the documents in this document store – use the query string to filter the results returned. Also accepts document submissions.

Query String Options

The query string defines the filters to apply to the document list. It follows url and takes the form:

```
?a[bbbbbb]=cccccc
```

Where:

a = query command (see below for query command types)

b = key name

c = filter value

Multiple filters can be specified, separated by an ampersand (&):

```
?a[bbbbbb]=cccccc&d[eeeeee]=ffffff
```

E.g.

```
https://SERVER:6443/docStore/store/store1/document/?q[bbbbbb]=cccccc&q[dddddd]=eeeeee
```

Will return a list of documents which have the value ‘cccccc’ in key ‘bbbbbb’ and the value ‘eeeeee’ in key ‘dddddd’

Query Command Types

q – Exact Match

https://SERVER:6443/docStore/store/store1/document/?q[xxxxx]=yyyyy
which will return all documents which have the value yyyyy in the key xxxxx.

(SQL WHERE xxxxx='yyyyy')

Q – Exact String Match, Case Insensitive

https://SERVER:6443/docStore/store/store1/document/?Q[xxxxx]=yyyyy
This will return all documents which have the value yyyyy (in any case) in the key xxxxx.

(SQL WHERE UCASE(xxxxx)='YYYYY')

l – Containing String Match

https://SERVER:6443/docStore/store/store1/document/?L[xxxxx]=yyyyy
This will return all documents which have the value yyyyy anywhere in the key xxxxx

(SQL xxxxx LIKE '%YYYYY%')

L – Containing String Match, Case Insensitive

https://SERVER:6443/docStore/store/store1/document/?L[xxxxx]=yyyyy
This will return all documents which have the value yyyyy (in any case) anywhere in the key xxxxx

(SQL UCASE(xxxxx) LIKE '%YYYYY%')

Range Queries

For Date and Numeric fields, it is possible to match a range of values:
https://SERVER:6443/docStore/store/Docstore%20Demo/document/?q[xxxxx]=2010-12-01%20to%202010-12-25
This will return all documents which have a date between 1st and 25th december 2010 in the key xxxxx

(SQL xxxxx BETWEEN '2010-12-01%20' and '202010-12-25')

https://SERVER:6443/docStore/store/Docstore%20Demo/document/?q[xxxxx]=0%20to%2099
This will return all documents which have a value between 0 and 99 in the key xxxxx

(SQL xxxxx BETWEEN 0 and 99)

Query Key Names

Key Name	Description	Examples
All Docstore Key names	All the Keys that have been defined in the Docstore	q[INVNUM]=1043125 return documents with a value of 1043125 for the key INVNUM The defined keys are listed at: http://SERVER:6443/config/docStore/store/STORENAME/docTypeKey/
docDescription	document Description	
docType	document Type	L[docType]=invoice (All documents that have a document type containing the word 'invoice')
docTypeID	document Type ID	
docID	document ID	q[docID]=854&q[docID]=800
docPath	file path	L[docPath]=QNTC (All documents stored on QNTC)
docFileName	File Name	
docOriginalFileName	Original File Name	L[docOriginalFileName]=file.PDF
docMimeType	Mime Type	q[docMimeType]=jpg&q[docMimeType]=pdf
docCreator	Document Creator	Q[docCreator]=dave
docTimeAdded		q[docTimeAdded]=2016-01-01 00:00:00 to 2017-01-01 00:00:00 q[docTimeAdded]=2016-09-28 14:20:00 to 2016-09-28 14:30:00
docTimeUpdated		
docTimeAccessed		
docAccessCount	docAccessCount	q[docAccessCount]=100 to 100000
docAutList	File Authorisation List	
docAutPublic	File Public Authority	l[docAutPublic]=W (Find all public writable files) q[docAutPublic]=*RWX

Sort Key and Sort Order

sortKey=KEYNAME	The Key name to sort the results by (Default is docTimeAdded)
sortOrder=ASC or DESC	Sort Ascending, or Sort Descending (Default is DESC)

E.g.

`https://SERVER:6443/docStore/store/store1/document/?q[bbbbbb]=cccccc&q[dddddd]=eeeeee&sortKey=dddddd&sortOrder=ASC`

Will return the records sorted by key 'dddddd' with the lowest values first

Note that only records that have a value for key 'dddddd' will be returned.

Pagination

Because of the large number of possible records that could be returned, only a subset are returned. By default the first 20 records are returned, but that can be adjusted using the start and count keys:

start=1	The index of the first result to return. The default of 1 is used if no value is provided.
count=20	The number of results to return. The default of 20 is used if no value is provided

E.g.

```
https://SERVER:6443/docStore/store/store1/document/?q[bbbbbb]=cccccc&q[dddddd]=eeeeee&start=21&count=50
```

Will return the 21st to 70th records

Alternative Data return format

It is possible to return all the documents that match the query parameters in a zip archive. This can be achieved by setting the Content-Type to application/zip in the http headers or by adding the following to the query string:

contentType=zip

E.g.

```
https://SERVER:6443/docStore/store/store1/document/?q[bbbbbb]=cccccc&q[dddddd]=eeeeee&contentType=zip
```

This request will return a zip archive containing all the documents that have a value ‘cccccc’ in key ‘bbbbbb’ and the value ‘eeeeee’ in key ‘dddddd’. Well, the first 20

Additional Key Values to return

It is possible to return addition key values in addition to the default metadata that Docstore returns by default by specifying the values parameter:

keys=bbbbbb,dddddd

E.g.

```
https://SERVER:6443/docStore/store/store1/document/?q[bbbbbb]=cccccc&q[dddddd]=eeeeee&keys=bbbbbb,dddddd
```

This request will return the default metadata plus the values for key bbbbbb and key dddddd for each document, if set.

Default Metadata values returned (if present):

- url
- authorised
- authorisedToDelete
- docID
- docTypeID
- docPath
- docFileName
- docOriginalFileName
- docMimeType
- docCreator
- docTimeAdded
- docTimeUpdated
- docTimeAccessed
- docAccessCount
- docDescription
- docAutList
- docAutPublic
- docTypeProxyAuthObjectPath
- docTypeHideUnauthorisedDocs
- docTypeAutList

Redirect to first document returned - ifl

It may be desirable for the request to be redirected to the url of the first returned document rather than returning the result list xml. This can be achieved by adding the following to the query string:

Ifl=true

(in fact ifl=anythingatall or ifl=false will have the same effect. That is, if the ifl parameter is present, the redirect will be activated)

For example, when querying by Invoice number, there should only be one document returned. Showing a document list containing just that one document would be less efficient than redirecting the query and displaying the document immediately.

E.g.

`https://SERVER:6443/docStore/store/store1/document/?q[bbbbbb]=cccccc&q[dddddd]=eeeeee&ifl=true`

This request will be redirected to the url of the first document that has a value ‘cccccc’ in key ‘bbbbbb’ and the value ‘eeeeee’ in key ‘dddddd’.

(Note: ifl = I’m Feeling Lucky)

Submitting a Document

This is the URL used to submit a document to the Docstore. The http method must be POST unless `treatGetAsPost=true`

Parameters	Description
document	type file, specifies the document data.
description	description for the document
docType docTypeID	One or the other is required
[keyname]	The values for the key
PARMxx to PARM20	Any additional parameters (for description entity replacement)
authorityOwner	
authorityPublic	
authorisationList	

```
<html>
  <body>
    <h1>Minimal Docstore Document upload form</h1>
    <p></p>
    <form action="http://SERVER:6400/docStore/store/DOCSTORE1/document/"
                enctype="multipart/form-data" method="post">
      Document:              <input type="file" name="document"><br />
      Description:           <input type="text" name="description"><br />
      Generic char key:      <input type="text" name="key1"><br />
      Generic char key(2):   <input type="text" name="key1"><br />
      TEXT 30 CHARS:         <input type="text" name="TEXT30"><br />
      an integer:            <input type="text" name="INT"><br />
      <input type="hidden" name="docType" value="DocType1">
      <input type="submit" value="Upload" name="upload">
    </form>
  </body>
</html>
```

A sample html form that would submit a document to the Docstore. The xml response would be like the sample output below:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreStoreRequest.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <filenameURLWorkaround>TRUE</filenameURLWorkaround>
  <docStore>
    <documentStoreRequest>
      <result>
       <code>200</code>
       <message>OK</message>
       <description></description>
      </result>
      <document url="http://SERVER:6400/docStore/store/DOCSTORE1/document/281/"
      authorised="TRUE">
        <docID>281</docID>
        <docTypeID>1</docTypeID>
        <docPath>/cppd/DocStore/DOCSTORE1/DocType1</docPath>
        <docFileName>DTPATJZRVQ.txt</docFileName>
        <docOriginalFileName>notes.txt</docOriginalFileName>
        <docMimeType>2</docMimeType>
        <docCreator>DANIEL</docCreator>
        <docTimeAdded>2012-08-07-13.48.59.226908</docTimeAdded>
        <docTimeUpdated>0001-01-01-00.00.00.000000</docTimeUpdated>
        <docTimeAccessed>0001-01-01-00.00.00.000000</docTimeAccessed>
        <docAccessCount>0</docAccessCount>
        <docDescription>test</docDescription>
        <keyValues>
          <key1>key1</key1>
          <key1>key</key1>
          <INT>1234</INT>
          <TEXT30>key3</TEXT30>
        </keyValues>
      </document>
    </documentStoreRequest>
  </docStore>
</CPPD>
```

Document Store Document

https://SERVER:6443/docStore/store/Docstore%20Demo/document/[docID]/
This is the url of the data contained in the stored document. A GET call to this url will return the document itself, a PDF file for example.

Delete

A DELETE http request or a GET specifying treatRequestAsDelete=true will delete the document if the user has the required authority.

Update

This is the URL used to update the attributes of the document. The http method must be POST unless
treatGetAsPost=true

Parameters
authorityOwner
authorityPublic
authorisationList
originalFileName
description
ANY KEY NAME

Only parameters that need to be changed need to be submitted. If a parameter is not specified, its value will not be changed.

If a key is specified with a blank value, all of its values will be deleted.

Example:
https://SERVER:6443/docStore/store/[STORENAME]/document/[docID]/?TEST1=2011-01-04&TEST1=2011-01-05&TEST2=3&TEST3=&description=A%20better%20Test%20document

This would update/set two values for Key TEST1, update/set one value for key TEST2 and remove all values for key TEST3. Any values for other keys would be left alone. It would also set the document description to “A better Test document”

Document Store Document Metadata

https://SERVER:6443/docStore/store/Docstore%20Demo/document/[docID]/meta/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreDocumentMeta.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <document url="http://SERVER:6400/docStore/store/DocStore%20Demo/document/38/">
        <docID>38</docID>
        <docTypeID>2</docTypeID>
        <docPath>/cppd/DocStore/DocStore Demo/Supplier Invoices</docPath>
        <docFileName>DTMZPX0F00.pdf</docFileName>
        <docOriginalFileName>Supplier Invoice - 22444553
(P000062).pdf</docOriginalFileName>
        <docMimeType>1</docMimeType>
        <docCreator>DANIEL</docCreator>
        <docTimeAdded>2011-05-23-16.55.45.966216</docTimeAdded>
        <docTimeUpdated>0001-01-01-00.00.00.000000</docTimeUpdated>
        <docTimeAccessed>2011-06-06-17.33.46.793636</docTimeAccessed>
        <docAccessCount>2</docAccessCount>
        <docDescription>Supplier Invoice 22444553 From Demonstration
Company</docDescription>
        <keyValues>
          <APINVDT>2011-05-20</APINVDT>
          <APINVNO>22444553</APINVNO>
          <PONo>P000062</PONo>
          <SUPNAME>Demonstration Company</SUPNAME>
        </keyValues>
      </document>
      <documentType>
        <id>2</id>
        <name>Supplier Invoices</name>
        <description>Supplier Invoices (Accounts Payable)</description>
        <rootFolder>/cppd/DocStore/DocStore Demo/Supplier Invoices</rootFolder>
        <autoName>TRUE</autoName>
        <lifeSpan>0</lifeSpan>
        <proxyAuthorityObjectPath></proxyAuthorityObjectPath>
        <documentHandlingUsername></documentHandlingUsername>
        <docTypeKeys>
          <docTypeKey>
            <id>4</id>
            <name>APINVNO</name>
            <description>Supplier Invoice No.</description>
            <type>CHAR</type>
            <size>10</size>
            <scale>0</scale>
            <index>10</index>
            <required>TRUE</required>
            <unique>FALSE</unique>
          </docTypeKey>
          …
        </docTypeKeys>
      </documentType>
    </store>
  </docStore>
</CPPD>
```

Lists all the information known about this document.

Document Store Document Keys/Values Data

https://SERVER:6443/docStore/store/Docstore%20Demo/document/[docID]/key/

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreDocumentKeys.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <filenameURLWorkaround>TRUE</filenameURLWorkaround>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <document url="http://m15:6400/docStore/store/DocStore%20Demo/document/35">
        <keyValues>
          <PONo>124572</PONo>
          <SUPACC>S000001</SUPACC>
          <SUPNAME>ABC SUPPLIES LTD</SUPNAME>
        </keyValues>
      </document>
      <documentType>
        <id>3</id>
        <name>Purchase Orders</name>
        <description>Purchase Orders</description>
        <rootFolder>/cppd/DocStore/Inforum Demo/Purchase
        Orders</rootFolder>
        <autoName>TRUE</autoName>
        <lifeSpan>0</lifeSpan>
        <proxyAuthorityObjectPath></proxyAuthorityObjectPath>
        <documentHandlingUsername></documentHandlingUsername>
        <docTypeKeys>
          <docTypeKey>
            <id>7</id>
            <name>PONo</name>
            <description>Purchase Order No.</description>
            <type>CHAR</type>
            <size>10</size>
            <scale>0</scale>
            <index>10</index>
            <required>TRUE</required>
            <unique>FALSE</unique>
          </docTypeKey>
          ........
        </docTypeKeys>
      </documentType>
    </store>
  </docStore>
</CPPD>
```

GET: Lists all the key/value pairs for this document.

POST: updates the key/value pairs for this document

Parameters: [keyName]=[keyValue]

Notes: Multiple values can be specified for a key

Eg: [keyName]=keyValue1& [keyName]=keyValue2

v2.94 and above: existing values are updated, excess values removed, additional values added.

Pre v2.94: All existing values will be removed before the new values are added

To delete all values for a key, specify it with a blank value

Eg: [keyName]=

Authority: The user must have Write authority to the document

Document Store Document Key/Value Data

https://SERVER:6443/docStore/store/Docstore%20Demo/document/[docID]/key/[keyName]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/files/docStoreDocumentKeys.xsl"?>
<CPPD>
  <Library>CPPD</Library>
  <filenameURLWorkaround>TRUE</filenameURLWorkaround>
  <docStore>
    <store>
      <name>DocStore Demo</name>
      <document url="http://ml5:6400/docStore/store/DocStore%20Demo/document/35/key/">
        <keyValues>
          <PONo>124572</PONo>
        </keyValues>
      </document>
      <documentType>
        <id>3</id>
        <name>Purchase Orders</name>
        <description>Purchase Orders</description>
        <rootFolder>/cppd/DocStore/Inforum Demo/Purchase
        Orders</rootFolder>
        <autoName>TRUE</autoName>
        <lifeSpan>0</lifeSpan>
        <proxyAuthorityObjectPath></proxyAuthorityObjectPath>
        <documentHandlingUsername></documentHandlingUsername>
        <docTypeKeys>
          <docTypeKey>
            <id>7</id>
            <name>PONo</name>
            <description>Purchase Order No.</description>
            <type>CHAR</type>
            <size>10</size>
            <scale>0</scale>
            <index>10</index>
            <required>TRUE</required>
            <unique>FALSE</unique>
          </docTypeKey>
          ........
        </docTypeKeys>
      </documentType>
    </store>
  </docStore>
</CPPD>
```

GET: Returns information about this key for this document

POST: updates this key for this document

Parameters: keyValue=[keyValue]

Notes: Multiple values can be specified for a key

E.g: keyValue=keyValue1& keyValue =keyValue2

Any existing values for the key will updated then any additional keys added. If there were more key values before, then the surplus will be deleted.

To delete all values for the key, specify it with a blank value

E.g: keyValue=

Authority: The user must have Write authority to the document

Restart the Web Service

POST to **http://localhost:6400/config/?restartServer=true**

Note, this will only restart the web service, not the CPPD subsystem.

Appendix

Encoding and Decoding Sample Code

percent encoding

in C#:

```
System.Uri.EscapeDataString("Cobwebb Communications=&");
System.Uri.UnescapeDataString("Cobwebb%20Communications%3d%26");
```

In Java:

```
// * Utility class for JavaScript compatible UTF-8 encoding and decoding.

//http://stackoverflow.com/questions/607176/java-equivalent-to-javascripts-encodeur
icomponent-that-produces-identical-outpu
    public static String encodeURIComponent(String s)
    {
        String result = null;

        try
        {
                result = URLEncoder.encode(s, "UTF-8")
                                .replaceAll("\\+", "%20")
                                .replaceAll("\\%21", "!")
                                .replaceAll("\\%27", "'")
                                .replaceAll("\\%28", "(")
                                .replaceAll("\\%29", ")")
                                .replaceAll("\\%7E", "~");
        }

        // This exception should never occur.
        catch (UnsupportedEncodingException e)
        {
                result = s;
        }

        return result;
    }

    public static String decodeURIComponent(String s)
    {
        if (s == null)
        {
                return null;
        }

        String result = null;

        try
        {
                result = URLDecoder.decode(s, "UTF-8");
        }

        // This exception should never occur.
        catch (UnsupportedEncodingException e)
        {
                result = s;
        }

        return result;
    }
```

Key Types

SMALLINT
INTEGER
BIGINT
CHAR
VARCHAR
TIMESTAMP
DATE
TIME
FLOAT
REAL
DOUBLE
DECIMAL
NUMERIC

Automating Docstore creation using scripts

It's possible to call Docstore URLs to create Docstores, Document Types, Keys and Document Type Keys. It should be possible to create a script to automate the creation of any Docstore setup.

This video walks through how this can be done using the excellent Postman app.

Get Postman App from here:
https://www.postman.com/

Download and Import this collection into Postman:
DocStore API Tests v3.postman_collection.json