

CHTTPD - URL Interface Specification

[CHTTPD - URL Interface Specification](#)

[Introduction](#)

[Note on HTTP Methods](#)

[Note on HTTPS](#)

[Note on Authentication](#)

[The username and password can be provided using HTTP authentication.](#)

[The username and password can be set in the cookies:](#)

[The username and password can be passed in the URL](#)

[Login URL](#)

[Note on Return Values](#)

[Eg: Success Response:](#)

[Eg: Failure Response:](#)

[Interface URLs](#)

[Docstore Interface](#)

[Message Interface](#)

[Message Queue Interface](#)

[Remote Procedure Call Interface](#)

[Data Interface](#)

[CHTTPD - Warehouse URL Interface Spec](#)

[JSON Output](#)

Introduction

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

Note on HTTP Methods

The HTTP protocol has 4 methods: GET, POST, PUT and DELETE. Most Webpage 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, CHTTPD 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.

Eg:

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

Note on HTTPS

HTTPS is available in CHTTPD v4.04 and above which was released with CPPD 8.2.80

By default HTTPS uses the port 6443, HTTP uses the port 6400

Note on Authentication

There are several ways to provide credential:

CHTTPD accepts IBM i user credentials: Either username and password or username and token should be provided. For more information on tokens, see the [note in the appendix](#).

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 token can be set in the cookies:

(From CHTTPD v4.75 and above)

`chttpd.user = USERNAME`

`chttpd.tokenHex = AABBCCDDEEFF001122334455667788...` (See Note 1)

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` (See Note 1)

or

`chttpd.user = USERNAME`

`chttpd.password = PASSWORD`

or

`CPPD = USERNAME|AABBCCDDEEFF0011` – this was the System21 Workspace Single signon method

Or

User = USERNAME

Pass = PASSWORD

The username and token can be passed in the URL

eg:

<https://SERVER:6443/diagnostics/?user=USERNAME&token=AABBCCDDEEFF...>

The username and password can be passed in the URL

eg:

<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](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>

Note on 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']")
```

Eg: 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>
```

For JSON Requests

```
{
  "cppd": {
    "response": {
      "code": 200,
      "message": "OK",
      "description": "Document Keys successfully updated from document 256"
    }
  }
}
```

Eg: 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>
```

The human readable error message can be retrieved from the html using `id="statusDescription"`, eg in Android:

```
public static String parseChttpdResponse(InputStream stream)
{
    String sMsg = "";
    try
    {
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        DocumentBuilder db = dbf.newDocumentBuilder();
        Document document = db.parse(stream);
        Element p = document.getElementById("statusDescription");
        sMsg = getNodeText(p);
    }
    catch(Exception e)
    {
        e.printStackTrace();
        Log.e(TAG + " parseChttpdResponse", "Exception : " + e.getMessage(), e);
    }
    return(sMsg);
}
```

For JSON Requests

```
{
  "cppd": {
    "response": {
      "code": 500,
      "message": "Internal Server Error",
      "description": "Failed to insert new keys"
    }
  }
}
```

Interface URLs

Docstore Interface

Base URL: <https://SERVER:6443/docStore/>

Documentation: [CHTTPD - Docstore URL Interface Spec](#)

Provides a comprehensive interface to the Docstore document management system.

Message Interface

Base URL: <https://SERVER:6443/message/>

Documentation: [CHTTPD - Message URL Interface Spec](#)

Interface to send an email message via the Cobwebb server

Message Queue Interface

Base URL: <https://SERVER:6443/queue/>

A comprehensive interface to the Cobwebb server message queue, used for monitoring the status of outgoing emails and faxes.

Remote Procedure Call Interface

Base URL: <https://SERVER:6443/rpc/>

Documentation: [CHTTPD - Remote Procedure Call URL Interface Specification](#)

Interface to call programs on the server

Data Interface

Base URL: <https://SERVER:6443/data/>

Documentation: [CHTTPD - Data URL Interface Specification](#)

Interface to read and write database tables on the server

CHTTPD - Warehouse URL Interface Spec

Base URL: <https://SERVER:6443/erp/warehouse/>

Documentation: [CHTTPD - Warehouse URL Interface Spec](#)

Interface to read and write data to a warehouse application

Miscellaneous Interfaces

/login

/files

/config

/diagnostics

JSON Output

CHTTPD defaults to outputting XML but can be switched to JSON output by setting the Content-Type to [application/json](#) or adding &contentType=json to the query parameters.

JSON sample:

```

{
  "cppd": {
    "warehousing": {
      "pick": {
        "-id": "0000000100",
        "CONO01": "TN",
        "ORTP01": "1",
        "ORDN01": "00000001",
        "DESN01": "0",
        "USER01": "KIM",
        "PSTS01": "C",
        "PSTD01": "2017-02-23",
        "PSTT01": "14.41.00",
        "PEND01": "2017-02-24",
        "PENT01": "15.55.47",
        "PCFD01": "0001-01-01",
        "PCFT01": "00.00.00",
        "NLAB01": "0",
        "items": {
          "item": [
            {
              "CONO02": "TN",
              "ORTP02": "1",
              "ORDN02": "00000001",
              "DESN02": "0",
              "BINN02": "BIN1",
              "LOCD02": "SR",
              "ORDL02": "1",
              "CATN02": "PRODUCT1",
              "OQTY02": "10.000",
              "PQTY02": "75.000",
              "PDES02": "New Car fo Dan"
            },
            {
              "CONO02": "TN",
              "ORTP02": "1",
              "ORDN02": "00000001",
              "DESN02": "0",
              "BINN02": "BIN2",
              "LOCD02": "SR",
              "ORDL02": "2",
              "CATN02": "PRODUCT2",
              "OQTY02": ".500",
              "PQTY02": ".000",
              "PDES02": "New Car for Barny"
            }
          ]
        }
      }
    }
  }
}

```

XML Sample:

```
<?xml version="1.0" encoding="utf-8"?>
<cppd><warehousing><pick id="000000100">
  <CONO01>TN</CONO01>
  <ORTP01>1</ORTP01>
  <ORDN01>0000001</ORDN01>
  <DESN01>0</DESN01>
  <USER01>KIM</USER01>
  <PSTS01>C</PSTS01>
  <PSTD01>2017-02-23</PSTD01>
  <PSTT01>14.41.00</PSTT01>
  <PEND01>2017-02-24</PEND01>
  <PENT01>15.55.47</PENT01>
  <PCFD01>0001-01-01</PCFD01>
  <PCFT01>00.00.00</PCFT01>
  <NLAB01>0</NLAB01>
  <items>
    <item>
      <CONO02>TN</CONO02>
      <ORTP02>1</ORTP02>
      <ORDN02>0000001</ORDN02>
      <DESN02>0</DESN02>
      <BINN02>BIN1</BINN02>
      <LOCD02>SR</LOCD02>
      <ORDL02>1</ORDL02>
      <CATN02>PRODUCT1</CATN02>
      <OQTY02>10.000</OQTY02>
      <PQTY02>75.000</PQTY02>
      <PDES02>New Car fo Dan</PDES02>
    </item>
    <item>
      <CONO02>TN</CONO02>
      <ORTP02>1</ORTP02>
      <ORDN02>0000001</ORDN02>
      <DESN02>0</DESN02>
      <BINN02>BIN2</BINN02>
      <LOCD02>SR</LOCD02>
      <ORDL02>2</ORDL02>
      <CATN02>PRODUCT2</CATN02>
      <OQTY02>.500</OQTY02>
      <PQTY02>.000</PQTY02>
      <PDES02>New Car for Barny</PDES02>
    </item>
  </items>
</pick></warehousing></cppd>
```

Appendix

Tokens

The Docstore web service (chttpd) has been enhanced (in v4.75 and above) to validate a user when IBM i tokens are provided as an alternative to passwords.

Option 1: pass in username and password (user=DANIEL&password=*****)

Option 2: pass in username and an IBM i token. (user=DANIEL&token=ABCDEFGG...)

If Infor System i Workspace Anywhere has been [configured to use the IBM i Token Generator](#) then it is possible to get it to pass the username and IBM i token when opening a docstore link. This means the user is transparently signed into Docstore.

Configuring the IBM i Token Generator

(Note: The latest versions of all these guides can be found on Knowledge Base article 1963350 on the Infor Support Portal.)

[System i Workspace AnyWhere - Installation & Admin Guide.pdf](#)

P162: Installing and Configuring the IBM i Token Generator - Walks through setting up the generator.

[System i Workspace AnyWhere - Product Guide.pdf](#)

P155: Chapter 6 Substitution parameters

describes the %user% and %pass% substitution values that can be used when constructing a new tab url.

“If Use System i user token for Passwords is enabled for the current Infor System i Workspace AnyWhere Profile, then the %pass% value will contain an IBM i Profile Token, otherwise it will contain an unencrypted password.”