**WebHDFS**

<http://www.informit.com/articles/article.aspx?p=2460260&seqNum=4>

As long as an application needs to access data stored in HDFS **from inside a cluster** or another machine **on the network**, it can use a high-performance **native** **protocol** or native Java API and be fine. But what if an external application wants to access or manage files in the HDFS over the Internet or HTTP or the Web?

For these kinds of requirements, an additional protocol was developed. This protocol, called **WebHDFS**, is based on an industry-standard RESTful mechanism that does not require Java binding. It works with operations such as reading files, writing to files, making directories, changing permissions, and renaming. It defines a public HTTP REST API, which permits clients to access HDFS over the Web**. Clients can use common tools such as curl/wget to access the HDFS**.

WebHDFS provides web services access to data stored in HDFS. At the same time, it retains the security the native Hadoop protocol offers and uses parallelism, for better throughput.

To enable WebHDFS (REST API) in the name node and data nodes, you must set the value of dfs.webhdfs.enabled configuration property to **true** in hdfs-site.xml configuration file as shown in the [Figure 3.15](javascript:popUp('/content/images/chap3_9780672337277/elementLinks/03fig15_alt.jpg')).

[](javascript:popUp('/content/images/chap3_9780672337277/elementLinks/03fig15_alt.jpg'))

[FIGURE 3.15](javascript:popUp('/content/images/chap3_9780672337277/elementLinks/03fig15_alt.jpg')) WebHDFS-related configuration.

------------------------------------------------------------------------ |

<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/WebHDFS.html>

##### MapReduce REST APIs

##### YARN REST APIs

##### [WebHDFS REST API](https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/WebHDFS.html) : <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/WebHDFS.html>

<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/WebHDFS.html#FileSystem_URIs_vs_HTTP_URLs>

A generic URI is of the form: **scheme**:[**//**[**user:password@**]**host**[**:port**]][**/**]**path**[?**query**][#**fragment**]

 Scheme : popular schemes include [http](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol), [ftp](https://en.wikipedia.org/wiki/File_Transfer_Protocol), [mailto](https://en.wikipedia.org/wiki/Mailto), [file](https://en.wikipedia.org/wiki/File_URI_scheme), and [data](https://en.wikipedia.org/wiki/Data_URI_scheme)

**authority :** [**user:password@**]**host**[**:port**] , Host  is [hostname](https://en.wikipedia.org/wiki/Hostname), or an [IP address](https://en.wikipedia.org/wiki/IP_address).

**path**, which contains data, usually organized in hierarchical form, that appears as a sequence of segments separated by slashes

optional **query**, separated from the preceding part by a question mark (?), .

hierarchical part

┌───────────────────┴─────────────────────┐

authority path

┌───────────────┴───────────────┐┌───┴────┐

abc://username:password@example.com:123/path/data?key=value#fragid1

└┬┘ └───────┬───────┘ └────┬────┘ └┬┘ └───┬───┘ └──┬──┘

scheme user information host port query fragment

Reference: <https://en.wikipedia.org/wiki/Uniform_Resource_Identifier>

File System URI of Webhdfs - webhdfs://<HOST>:<HTTP\_PORT>/<PATH> , i.e FileSystem scheme of webhdfs is **webhdfs://**

Corresponding HDFS URI for the above WebHdfs URI is - hdfs://<HOST>:<RPC\_PORT>/<PATH>

**In the REST API**, the **prefix** **“/webhdfs/v1”** is inserted in the path and a query is appended at the end.

http://<HOST>:<HTTP\_PORT>/webhdfs/v1/<**PATH**>?op=...

|  |  |
| --- | --- |
| dfs.webhdfs.enabled | Enable/disable WebHDFS in Namenodes and Datanodes |

## File and Directory Operations:

### Create and Write to a File