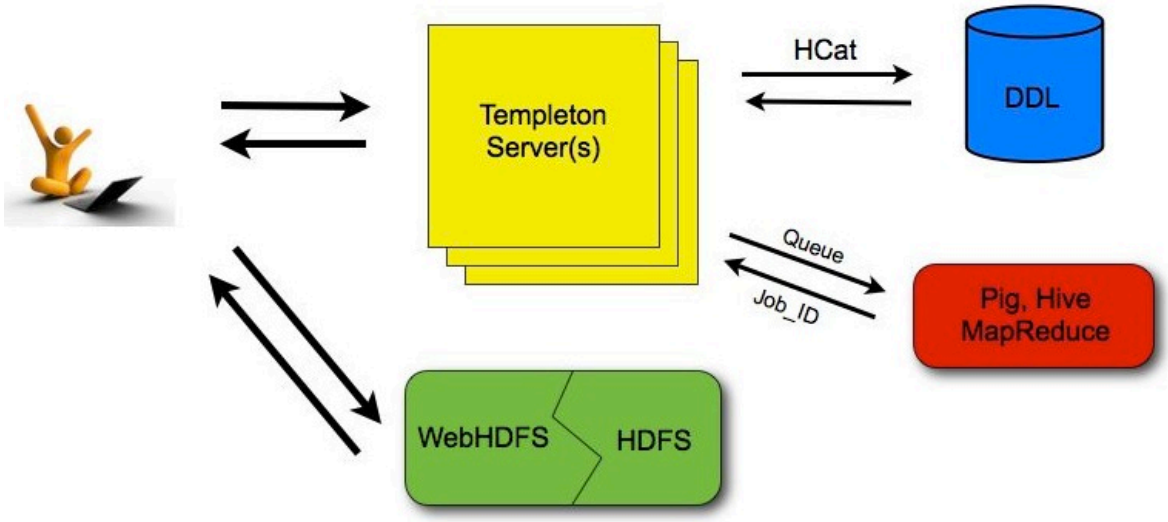
HCatalog REST API

Table of contents

1 Introduction

This document describes HCatalog REST API. As shown in the figure below, developers make HTTP requests to access Hadoop MapReduce, Pig, Hive, and HCatalog DDL from within applications. Data and code used by this API is maintained in HDFS. HCatalog DDL commands are executed directly when requested. MapReduce, Pig, and Hive jobs are placed in queue by and can be monitored for progress or stopped as required. Developers specify a location in HDFS into which Pig, Hive, and MapReduce results should be placed.



2 URL format

HCatalog's REST resources are accessed using the following URL format:

`http://`*`yourserver`*`/templeton/v1/`*`resource`*

where "*yourserver*" is replaced with your server name, and "*resource*" is replaced by the HCatalog resource name.

For example, to check if the server is running you could access the following URL:

```
http://www.myserver.com/templeton/v1/status
```

3 Security

The current version supports two types of security:

- Default security (without additional authentication)
- Authentication via Kerberos

3.1 Standard Parameters

Every REST resource can accept the following parameters to aid in authentication:

- user.name: The user name as a string. Only valid when using default security.
- SPNEGO credentials: When running with Kerberos authentication.

3.2 Security Error Response

If the user.name parameter is not supplied when required, the following error will be returned:

```
{
  "error": "No user found.  Missing user.name parameter."
}
```

4 WebHDFS and Code Push

Data and code that are used by HCatalog's REST resources must first be placed in Hadoop. When placing files into HDFS is required you can use whatever method is most convienient for you. We suggest WebHDFS since it provides a REST interface for moving files into and out of HDFS.

5 Error Codes and Responses

The server returns the following HTTP status codes.

- **200 OK:** Success!
- **400 Bad Request:** The request was invalid.
- **401 Unauthorized:** Credentials were missing or incorrect.
- **404 Not Found:** The URI requested is invalid or the resource requested does not exist.
- **500 Internal Server Error:** We received an unexpected result.
- **503 Busy, please retry:** The server is busy.

Other data returned directly by the server is returned in JSON format. JSON responses are limited to 1MB in size. Responses over this limit must be stored into HDFS using provided options instead of being directly returned. If an HCatalog DDL command might return results greater than 1MB, it's suggested that a corresponding Hive request be executed instead.

6 Log Files

The server creates three log files when in operation:

- **templeton.log** is the log4j log. This the main log the application writes to.
- **templeton-console.log** is what Java writes to stdout when the server is started. It is a small amount of data, similar to "hcat.out".

- **tempelton-console-error.log** is what Java writes to stderr, similar to "hcat.err".

In the tempelton-log4j.properties file you can set the location of these logs using the variable templeton.log.dir. This log4j.properties file is set in the server startup script.

7 Project Name

The original work to add REST APIs to HCatalog was called Templeton. For backward compatibility the name still appears in URLs, log file names, etc. The Templeton name is taken from a character in the award-winning children's novel Charlotte's Web, by E. B. White. The novel's protagonist is a pig named Wilber. Templeton is a rat who helps Wilber by running errands and making deliveries as requested by Charlotte while spinning her web.