Templeton

Table of contents

1 Introduction

Templeton provides a REST-like web API for HCatalog and related Hadoop components. 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 Templeton is maintained in HDFS. HCatalog DDL commands are executed directly when requested. MapReduce, Pig, and Hive jobs are placed in queue by Templeton and can be monitored for progress or stopped as required. Developers specify a location in HDFS into which Templeton should place Pig, Hive, and MapReduce results.
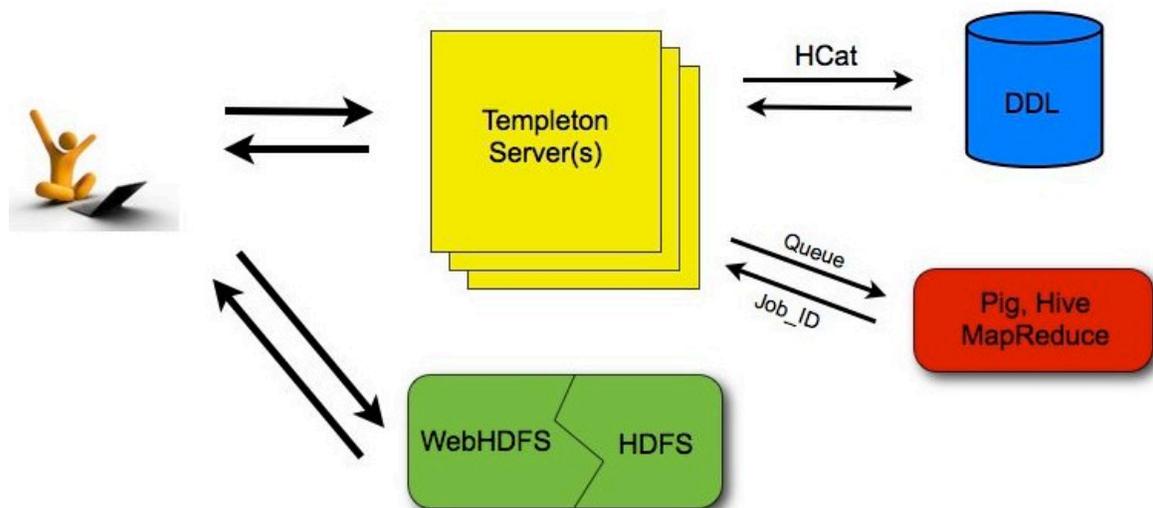
2 URL format

Templeton 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 Templeton resource name.

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

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

3 Security

The current version of Templeton supports two types of security:

- Default security (without additional authentication)
- Authentication via [Kerberos](#)

3.1 Standard Parameters

Every Templeton 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, Templeton returns the following error:

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

4 WebHDFS and Code Push

Data and code that are used by Templeton resources must first be placed in Hadoop. The current version of Templeton does not attempt to integrate or replace existing web interfaces that can perform this task, like [WebHDFS](#). (Integration of these functions in some way, perhaps forwarding, is planned for a future release.) When placing files into HDFS is required you can use whatever method is most convienient for you.

5 Error Codes and Responses

The Templeton 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 Templeton is currently 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.

Page 3

6 Log Files

The Templeton 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 Templeton 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 Templeton project is named after the 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.

Page 4