**Recipes Data Development and Design**

**Data Flow Model Doc**

Table of Contents

[Recipes Data flow Development Architecture Diagram: 3](#_Toc522088816)

[Development Phase: 4](#_Toc522088817)

[Step 1: Oozie Script creation: 4](#_Toc522088818)

[Oozie configuration: 4](#_Toc522088819)

[Transfer mechanics: 4](#_Toc522088820)

[Step2: Using SPARK with Scala context: 4](#_Toc522088821)

[Step 3: Hive table creation, definitions and mapping: 4](#_Toc522088822)

[Create the hive table schema by using the below code: 5](#_Toc522088823)

[Recipes Hive Table: 5](#_Toc522088824)

[Step 4: Software Testing Phase: 5](#_Toc522088825)

[ Functionality testing: 5](#_Toc522088826)

[ Performance testing: 5](#_Toc522088827)

[Step5: Version control 5](#_Toc522088828)

# Recipes Data flow Development Architecture Diagram:

**Recipes Servers**

**SFTP PUSH** OOZIE script

**CSV files on**

**edge node**

**Read the CSV Files**

**SPARK-SCALA**

**Load Data**

**Recipes Hive Table**

**Business Marketing**

# Development Phase:

## Step 1: Oozie Script creation:

Custom Oozie script will be running on the edge node.

An Oozie script can be developed and scheduled for automation of the data flow from the edge node to the Hive tables.

The frequency of the scheduler is 1hr for each run.

### Oozie configuration:

1. Job.properties
2. Workflow.xml
3. Beeline\_runner.sh
4. Coordinator.xml

Transfer mechanics:Recipes server does sftp push 1 csv file every hour onto edge node with 1TB data volume. This csv file will be processed as a single in-memory unit by using Spark/Scala and load into Hive table and perform the business analytics.

## Step2: Using SPARK with Scala context:

Spark – Scala technology (data framework) will be used for reading the CSV file, parsing and extracting the data and loading to Recipes Hive table as defined in the above architecture diagram.

The incorrect format data can be loaded to Error Hive table.

## Step 3: Hive table creation, definitions and mapping:

For accessing hive use the below command on the edge node:

**beeline -u "jdbc:hive2://HOSTNAME:10000/default;principal=hive/HOST@Domain;transportMode=http;httpPath=cliservice" -p "" -n ""**

### 

### Create the hive table schema by using the below code:

create table data\_table (

recipe\_id string,

recipe\_name string,

description string,

ingredient string,

active string,

updated\_date timestamp,

created\_date timestamp

) ;

### Recipes Hive Table:

The field data types and hive table definitions, description and mapping for this hive table are defined in the attached requirement word document.

## Step 4: Software Testing Phase:

* Functionality testing: The functionality of the product can be tested and the queries provided by the business will be performed on the developed hive table to ensure that the results are populated appropriately for reporting purpose. The test cases are also created to cover the end to end functionality of the application.
* Performance testing:The query run time can be captured and there is a possibility of the code normalization to increase the performance of the run time or to create the index on top of a specified column.

## Step5: Version control

* GIT HUB