Dash View

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| V 1.0 | Cheng Yi | 9/10/2016 |  |
| V 1.1 | .. | 9/16/2016 | Add First Release Requirement |
| V 1.2 | .. | 10/10/2016 | Add Flow Requirement |
| V 1.3 | .. | 10/13/2016 |  |

# TOC

Contents

[1. Document History 1](#_Toc466004262)

[2. TOC 1](#_Toc466004263)

[3. Technology 2](#_Toc466004264)

[4. Functional Modules 3](#_Toc466004265)

[4.1. Account Management and Editor 3](#_Toc466004266)

[4.1.1. Account 3](#_Toc466004267)

[4.1.2. Group 4](#_Toc466004268)

[4.1.3. Group Permission 4](#_Toc466004269)

[4.1.4. Project 4](#_Toc466004270)

[4.2. Data Editor 4](#_Toc466004271)

[4.2.1. Data Schema Editor 4](#_Toc466004272)

[4.2.2. Data Viewer 5](#_Toc466004273)

[4.3. Flow Management 5](#_Toc466004274)

[4.3.1. Flow Definition 5](#_Toc466004275)

[4.3.2. Operations 5](#_Toc466004276)

[4.4. Flow Editor 6](#_Toc466004277)

[4.4.1. Edit Flow 6](#_Toc466004278)

[4.4.2. Edit Action Properties 6](#_Toc466004279)

[4.4.3. Edit Data Source 6](#_Toc466004280)

[4.4.4. Edit Data Schema 6](#_Toc466004281)

[4.5. Flow Tester 6](#_Toc466004282)

[4.5.1. Run Action 6](#_Toc466004283)

[4.5.2. Execute Flow 7](#_Toc466004284)

[4.5.3. View Data 7](#_Toc466004285)

[4.5.4. View Log 7](#_Toc466004286)

[4.5.5. Resume Flow 7](#_Toc466004287)

[4.6. Tableau Integration 7](#_Toc466004288)

[4.6.1. User Mapping 7](#_Toc466004289)

[4.6.2. Project Mapping 7](#_Toc466004290)

[4.6.3. Display Worksheet 7](#_Toc466004291)

[4.6.4. Operations 7](#_Toc466004292)

[5. Custom DashView 7](#_Toc466004293)

[5.1. Data Source 7](#_Toc466004294)

[5.2. Query 7](#_Toc466004295)

[5.3. View Type 8](#_Toc466004296)

[5.3.1. Table View 8](#_Toc466004297)

[5.3.2. Map View 8](#_Toc466004298)

[5.4. Execution 8](#_Toc466004299)

[6. Reference 8](#_Toc466004300)

# Technology

1. Using real time (async), modular java script technology.
2. Using npm to manage the javascript library
3. Javascript library jquery, angular js and D3 for svg and canvas
4. Based mainly on open source projects

# Functional Modules

## Account Management and Editor



Figure 1 Account Mgmt Data Model

### Account

#### Definition

* Username
* Password
* Email
* GroupId
* UpdateTime

#### Operations

### Group

#### Definiton

* GroupId
* UpdateTime

#### Operations

### Group Permission

#### Definition

Is defined as actions permitted on a set of projects

For example: An account can have

Read on Project1, Project 2

Write on Project 3, Project 4

This is a relationship entity

* GroupId
* Action
* ProjectId
* UpdateTime

#### Operations

### Project

#### Project Definiton

* ProjectId
* UpdateTime
* Project Content (json definition)
  + A set of workflow (ETL, analysis)
  + A set of dataset (on the hdfs)
  + A corresponding tableau project.

Or

* + A set of Custom DashView dashboards

#### Operations

## Data Editor

This will be used in the flow editor as well.

### Data Schema Editor

A data schema contains multiple table definition:

For each table there are table name and a list of fields.

Each field contains the field name and field type.

### Data Viewer

Data can be fetch via the manager API, can be from hdfs, etc.

Display the data in table format, row by row

## Flow Management

### Flow Definition

* Name
* UpdateTime
* FlowContent (json definition)
  + A set of nodes
  + A set of links
  + A set of datasets

### Operations

* create()
* update()
* delete()
* get()

## Flow Editor



Figure 2 Flow Editor

### Edit Flow

* Specify the flow properties
* Create Action Node
* Edit Links

### Edit Action Properties

* Specify action properties

### Edit Data Source

### Edit Data Schema

See Data Management

## Flow Tester

### Run Action

* User can run a specific action from the workflow after specifying the input data

### Execute Flow

* User can run the whole flow from Web

### View Data

* User can view the data for each Action (both input and output data sets, see Data Management)

### View Log

* User can view the detailed log for each Acton.

### Resume Flow

* User can resume the flow at any step by specifying the instanceId (which is the dataset).

## Tableau Integration

### User Mapping

* Map the user to tableau user

### Project Mapping

* Map the reports in the tableau project to current project (using tableau restful API)

### Display Worksheet

* Passing parameters to tableau dashboards using tableau javascript

### Operations

getAllTableauProjects()

getWorksheetsByProject()

# Custom DashView

## Data Source

1. Support JDBC Data Source
   1. Vertica
   2. Hive
2. Support Realtime Streaming Data Source
   1. Kafka

## Query

1. User can define measures and dimensions.
2. User can select the fields.
3. User can specify the filter
4. For the fields are selected for the query but not displayed means group by these fields.
5. Calculated fields

## View Type

### Table View

1. User can define the rows and columns

### Map View

For location data.

1. User can zoom in and zoo out

## Execution

1. Support Pull type, pull the data from data source.
2. Support Streaming data pushed to us, and refresh the corresponding view.

# Reference

1. Tableau javascript API: <https://community.tableau.com/community/developers/javascript-api>
2. <https://d3js.org/>
3. <http://freeboard.github.io/freeboard/>