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 |  |

# 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



Figure 1 Account Mgmt Data Model

### Account Definition

* Username
* Password
* Email
* GroupId
* UpdateTime

### Group Definition

* GroupId
* UpdateTime

### 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

### Project Definition

* 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

### Single Sign on

Account LDAP Server Integration

## 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

1. Edit Flow
2. Edit Action Properties
3. Edit Data Source

### Flow Tester

1. User can run the flow from Web
2. User can check the data for each Action
3. User can check the log from the Web
4. User can resume the flow at any step by specifying the data

## Dashboard Management

### Tableau Integration

1. Map the user to tableau user
2. Map the reports in the tableau project to current project (using tableau restful API)
3. Passing parameters to tableau dashboards using tableau javascript

### Operation

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/>