

Probabilistic CAP implementation for Latency and Consistency SLAs

Google Chart For Visualization
&
Kong API Gateway

Design Specification
Version 0.1

Ambika Bohra
Ambika.bohra@sjsu.edu

Date: 07/27/2018
Advisor: Prof. Rakesh Ranjan

Version History
Introduction
References
Requirements
Functional Overview
Configuration/ External Interfaces
Debug
Logging
Counters
<u>Implementation</u>
Testing
General Approach
Unit Tests
Appendix

Version History

Version	Changes
1.0	Architectural Setup

Introduction

Google chart are a perfect solution for making visualization for all type of data. A vast range of ready-to-use charts can be plotted such as line chart, tree maps, histogram etc. The charts are based on SVG/HTML technologies to give a cross-browser compatibility and cross platform portability to other devices. These charts are interactive and can be easily added to the website. Charts are embedded as JavaScript classes, so it is easy to integrate it with the code by importing the certain libraries.

DataTable class provides various methods for filtering, sorting, and modifying the data. Data are populated from webpage, database or other data provider. Data source should comply with the chart tool datasource protocol. The simplest implementation of the library involves inheriting from a single class, implementing a member function and running the data source as a servlet within a servlet container.

We are using Google charts on our dashboard to show the performance based on the specified SLA for a target system. For this, we would use Google Visualization API. Using this API, any type of data is retrieved from local browser or supported data source and displayed in the dashboard. Data source can also be implemented as Visualization API, it enables any application to access the data. Data source format is manageable for reporting, analysis and other chart application. So data can be visualized and new functionality can be added to the application.

Kong API gateway is a form of middleware, which is used for computing clients and API-based application. We would integrate it with Apache Cassandra and Riak databases. The features of Kong API is authentication, logging, security, request/ response transformation, analytics &

monitoring and traffic control. Kong API features are provided by plugins such as OAuth 2:0 Authentication, SSL, transformation and request size limiting.

References

1. Google Charts. (n.d.). Retrieved from <https://developers.google.com/chart/interactive/docs/>
2. Introduction to the Data Source Library (n.d.). Retrieved from https://developers.google.com/chart/interactive/docs/dev/dsl_intro
3. Load the Libraries (n.d.). Retrieved from https://developers.google.com/chart/interactive/docs/basic_load_libs
4. Google Chart API (n.d.). Retrieved from https://en.wikipedia.org/wiki/Google_Chart_API
5. Google Visualization API Reference (n.d.). Retrieved from <https://developers.google.com/chart/interactive/docs/reference>
6. Kong installation (n.d.). Retrieved from https://docs.konghq.com/install/docker/?_ga=2.233969821.2013851789.1532647971-1838378268.1530755954

Requirements

Following are the requirements for Google Chart:

1. Initial setup of google chart API
2. Include google chart javascript file
3. Google Visualization API
4. Google Visualization Library for the Google Web Kit (GWT)

5. Integration from Kong API Gateway
6. Implement data source or get data from the application
7. Loading the libraries
8. Java version 1.8
9. Browser supporting SVG
10. Dashboard instance

Functional Overview

Following are the salient features of google chart:

- Open-source
- Compatibility to almost all the browsers and mobile platforms
- Light-weighted loader.js library
- Dynamic chart generation
- Date-time support
- External data can be loaded from server dynamically

The required libraries are loaded one time for multiple charts. For this, a call to `google.chart.load()` function is required for specific chart type. Google Chart Tools needed to be wrapped in a JavaScript class `google.visualization.DataTable`, which is defined in the Google Visualization library. The format of data is different for different types of charts. A datatable instance can be created by sending query to chart tools datasource. This returned DataTable can be copied, modified, or copied into a DataView the same as any other DataTable. Similarly, by

visualization API data can be accessed from creating the standard DataTable format and any data source that supports the API. The Visualization API allows a Javascript API for accessing the charts. It provides a documented wire protocol and a way for anyone to expose their data sources to any of the APIs visualizations. The API has a defined event model that allows charts to throw and receive events and thus communicate with the host page .

Configuration/ External Interface

- Browser should support SVG (chrome, firefox etc),
- Java Package
- Docker

Implementation

Following are the implementation steps for google chart setup:

- Create a HTML page with google chart libraries
- Load the Visualization API and the corechart package.
- Set a callback to run when the Google Visualization API is loaded.
- Populate the data by creating datatable instances.
- Integrate the google chart with the data collected from Kong API gateway (require a API)
- Create configuration
- Create a dashboard - add multiple charts
- Draw the chart

Following are the steps involved for Kong API gateway setup:

- Docker installation of Kong API
 - Docker network creation
 - Start the database
 - Prepare database
 - Start Kong

Testing

General Approach

For this design specification, basic testing is to checking if the data is integrated properly with the Kong API. Additionally, the data is populated to datatable instance and able to create couple of charts to show some statistics.

Unit Tests

- Charts should dynamically change as the data changes and it should response immediately.
- Check the graph with various filters and Values in Graph should be in Proper culture
- Test the Performance of Graph load with different amount of data
- Visualize simple statistics to show the performance based on the specified SLA

Appendix

Following are some sample configured charts:

