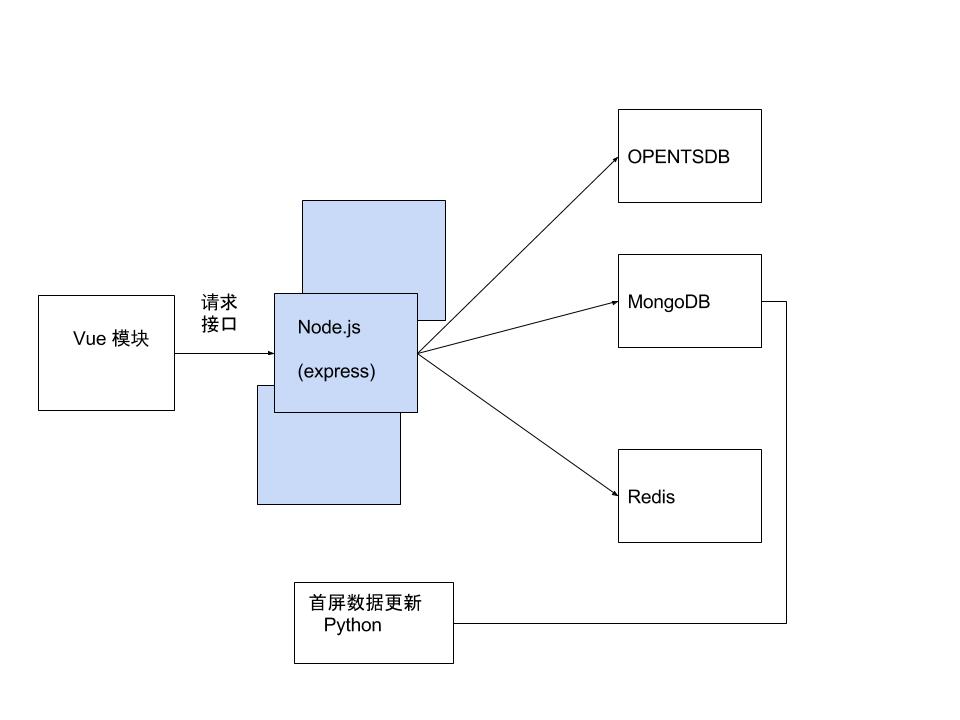
Web模块技术文档

1. 系统架构



图一. Web部分架构图

架构图如图一所示，总体分为三个模块:Vue模块、Node.js(express)后台模块、数据库模块。

1.Vue模块：把前端的js通过webpack打包到用户浏览器，由用户浏览器进行前端页面功能处理。

2.Node.js模块： 接收前端发送请求和转发前端请求到Opentsdb数据库，作用：记录系统中所有的视图信息、告警策略和告警历史信息和用户信息。

3.数据库模块：

Redis:用于实现微信与系统用户之间的绑定和session的存储

Mongodb： 存储视图、告警、用户等需要持久化存储的信息

Opentsdb： 用于提供数据返回给视图进行画图的操作

1. 功能模块讲解

##TODO 前端部分由东俊进行补充

1. Node部分
2. 视图参数的增删改查

因为我们系统中图上面可以配置的参数相对比较多（有单位，左右y轴，还有指标和每个特定的tag），因此采用了Mongodb的设计，可以把很多复杂的字段（用于表述图的参数，但不是用于查询或者读取的参数的地方），以文档的形式存放。

1. 告警策略的增删改查和告警历史的查询

对于 一个告警系统需要支持的是

1. 告警等级 （用于区分告警的严重程度）
2. 告警策略名 （用于区分告警规则）
3. 告警具体规则 （用于进行检测）
4. 告警通知的人 （用于告知需要处理告警的人）
5. 告警的时间（或者回复的时间），具体信息，和告警策略目前的状态（用于展示或者进行排错）

因此，我们利用了Mongodb的Document特性和Schma-less特性：

创建了一个以高表为雏形的Nosql表情况

把比较复杂的内容（如告警规则、告警通知组，告警历史）以文档的形式存在了主表的一个字段中进行保存（并且关联起来）。减少了多次的查询带来的对数据库的压力

告警策略告警用户的管理：

利用了Mongodb的schema-less的特性，把数据结构嵌套起来，存放到告警策略中，通过标识特定的用户名来表明此用户组中，这些特定的用户需要被告警通知。并且由于Schema-less的特性，对于我们继续做延展的功能开发上面变得更加简单，只需要插入新功能所需的字段即可进行新功能的开发，减少了每次表的变动，使得系统在快速开发和添加功能上面可以变得更加灵活

1. 用户的增删改查和用户组的增删改查

对于用户的设计和组方面的设计，都是利用了MongoDB本身的特性来进行规划。尽可能把不是经常用于查询的字段以文档的形式存放起来，而不去做跨表的查询，来减少对数据库的压力。Document的特性，我们可以把权限存放为一个字段，拿到字段后自己进行判断，而不用像用mysql那样，每个权限存储一张表，进行多张表的跨表查询来获取该用户应该拥有的权限

用户权限和组权限的管理：

通过用户与组权限相同进行控制，默认有admin组，admin组默认有root用户，拥有所有的权限（包括创建、删除用户、修改用户），如果想为用户升级权限或者降低权限，只需要绑定到一个新的组上面来进行。

1. 用户微信在手机端和pc端的绑定

Pc端上的绑定，请求有的微信模块获取公众号的二维码，然后从微信接口接收到返回信息后会进行redis的订阅发布，把用户的微信特征码发送到服务端这边，使得用户和用户微信特征可以进行绑定

通过在session中添加微信绑定的渠道字段，来进行用户和微信号之间的手机端绑定。

1. 会话管理

使用redis去保留用户的session，用户登陆成功后可以获取session\_id可以保持7天的登录状态

1. 转发请求到opentsdb进行查询
2. 跨域问题

使用了express社区上的cors跨域插件来解决特定请求可以进行跨域请求

1. 防SQL注入

9. 首页数据的更新

通过查询Python部分创建的首屏对应功能的表来获取最新的数据

1. Python首屏数据更新

读取系统中关于告警、和基础资源的状态信息，来进行首屏上面的数据的更新，存放到mongodb中，包括了健康度，集群的状态，告警趋势等。可以通过设定运行频率来定时更新首页的数据。