# 服务器代码分析



主要就是支持大量的数据源，数据库里有很多种数据，对应的资源的请求也很多。

另外有个调度系统，负责响应用户的不执行数据源的查询工作。

## 几个代码量较大的文件夹

### query\_runner

各种数据源，包括mysql、mongodb、druid、presto，具体见下图。



### models

数据库的表及对应Model类



### handlers

对客户端的各种URL数据请求响应，大部分都是通过继承BaseResource来处理，少量（admin，authentication，emded，organization，setup[创建admin]）是通过routes.route()处理。





## 其他文件夹

### authentication

登录、认证

### cli

自定义命令行

### destinations

事件通知的各种方式，比如邮件



### metrics

对数据库的查询的统计

### serializers

把Query、QueryResult、Dashboard、Visualization、Widget、Alert、Job等对象序列化

### settings

配置

### tasks

redash有三类进程，server、scheduler、worker：server负责url响应；worker可以有多个，负责具体的查询；scheduler负责调度。他们之间通过2个redis库来交换数据。具体工作使用了开源库rq。



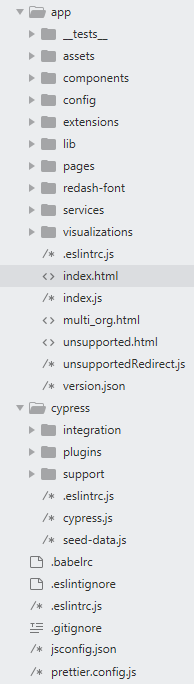
### templates

一些Jinja模板，用于生成html，比如邮件

### utils

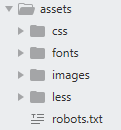
一些工具函数

# 客户端代码分析



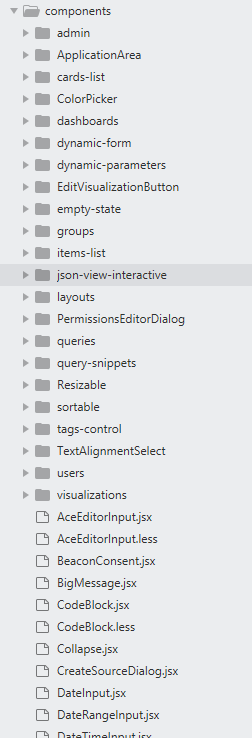
除去测试，总共4万多行，绝大部分都是react页面本身的jsx代码。

### app/asset，6117



图片、字体、CSS、LESS等资源。

### app/components，12285



JSX代码写的界面组件，页面的一部分，按功能分文件夹。组件产生html，组件本身可以作为标签被别的组件使用。VUE的组件又有同样的功能，组织结构可以参考。

### app/config

配置

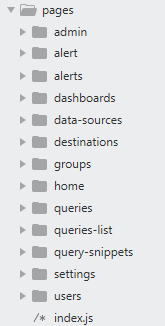
### app/extensions

空文件夹

### app/lib，398

少数几个JS文件，公用或工具性的函数。

### app/pages，8678

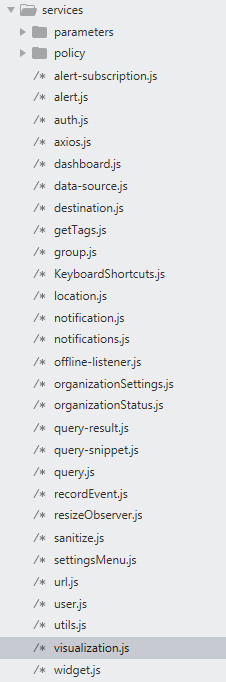


也是JSX代码写的界面组件，只不过指定了URL，对应一个url能看到的所有东西，所以就叫页面。

### app/redash-font

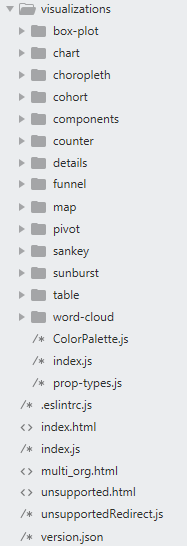
也是字体，LESS文件。

### app/services，3323



JS代码，3部分。一是parameters，redash运行在SQL语句中使用的可变参数；二是policy，客户端的用户权限检查；三是，服务器端有很多/api/开头的URL，对应各种数据的操作，这里对这种异步的HTTP请求进行封装，让界面使用起来就像是在调用普通函数。

### app/visualizations，11420



JSX代码写的把数据展示为图表，这些要用Echart重写。

### cypress

用cypress框架写的测试用例。