

# OpenDigger可视化大屏

License MIT

Node v15.14.0

English

## OpenDigger Big Screen



OpenDigger可视化大屏由 无监督学习队, 该项目是[OpenSODA](#)大赛的决赛作品

## 关于本项目

一个在线可访问, 可交互的可视化大屏. 数据来源于[X-lab2017/open-digger](#) 与 [Github API](#)

该项目由Vue支持, 在线访问地址: <http://open-digger.com/#/>

## 怎么查看OpenDigger 可视化大屏?

### 演示视频

<https://www.bilibili.com/video/BV1Nh4y1r7Gt/>

- 你只需要在浏览器输入地址 <http://open-digger.com/#/>, 即可在线访问

### Docker部署

1. 安装Docker: 请确保系统上已经安装了Docker. 他们可以按照官方文档中的说明进行安装: <https://docs.docker.com/get-docker/>

2. 拉取镜像: 你可以使用以下命令从Docker Hub上拉取上传的镜像

docker镜像地址: <https://hub.docker.com/repository/docker/beinvisibling/opendiggerbigscreen/general>

```
1 | docker pull beinvisibling/opendiggerbigscreen:latest
```

这将从Docker Hub下载并在他们的本地系统上存储您的镜像。

3. 运行容器：您可以使用以下命令来运行您的镜像：

```
1 | docker run -d -p 8081:8081 your-dockerhub-username/your-image-name
```

- `-d` 标志表示在后台运行容器。
- `-p 8080:3000` 标志将容器的端口 8081 映射到主机的端口 8081。这是假设您的Node.js应用程序在容器中使用的端口是8081。

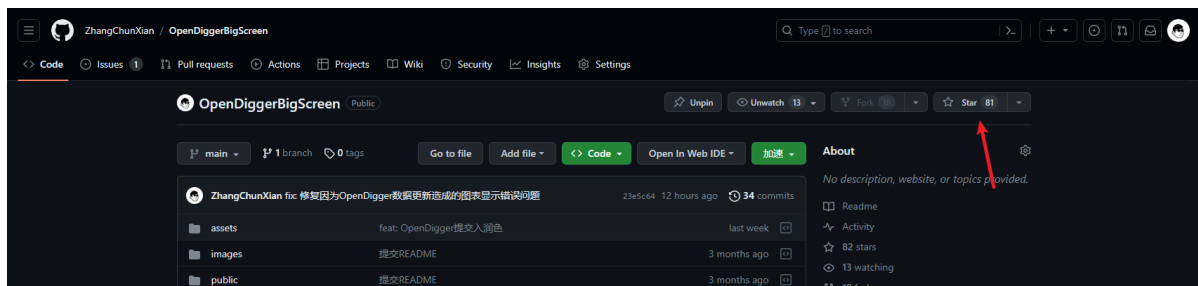
4. 访问应用程序：一旦容器正在运行，你就可以通过在浏览器中访问 `http://localhost:8081` 来访问您的Node.js应用程序。如果你在主机的其他端口上映射了容器端口，你需要相应地更改URL中的端口号。

这样，你就可以在他们的本地系统上运行和访问OpenDigger可视化大屏应用程序镜像了。可以使用 `docker stop container-id` 命令，其中 `container-id` 是容器的ID，可以使用 `docker ps` 命令查看。

## Github 仓库克隆

地址: <https://github.com/ZhangChunXian/OpenDiggerBigScreen>

得益于OpenDigger可视化大屏的高质量和本人在社交媒体上不遗余力的宣传, 目前star来到了81个!



- 克隆github仓库到本地

```
1 | git clone https://github.com/ZhangChunXian/OpenDiggerBigScreen.git
```

- npm 安装包, 推荐node版本为15.14.0

```
1 | cd OpenDiggerBigScreen
2 | npm install
```

- npm运行, 默认8080端口

```
1 | npm run serve
```

# OpenDigger可视化大屏介绍

## 场景与意义

世界需要开源, 开源需要被更多人看到, 要有更多人参与到开源中来. 本项目跳出开源一般意义上从代码的角度, 选择从更容易让大众理解和接收的视觉角度,吸引观众的注意力, 从可视化大屏角度去宣传开源, 更容易让普通人理解与接受. 因为普通人, 就算不懂代码与编程, 也能参与到开源中来.

## 实现方案

本大屏的数据来源于

- [X-lab2017](#)
- [Github API](#)

技术栈: Vue 2.6 Echarts 5.3 RestFul API

## 作品介绍

整个大屏的缩略图如下, 其中包含一些动态操作



## 作品特色

### 数据来源广泛

本大屏的数据来源于

- [X-lab2017](#)
- [Github API](#)

其中X-Lab2017中的API包含种类繁多的开源项目的各种信息, 信息如下

文件名称	描述
active_dates_and_times.json	项目每天的活跃度
activity_details.json	每人每天的活跃度
activity.json	项目每月的活跃度
attention.json	每月的关注度
bus_factor_detail.json	每人每月的巴士系数
bus_factor.json	项目每月的巴士系统
change_request_age.json	PR请求的生命时长（没有关闭的PR默认到23年3月份）
change_request_resolution_duration.json	PR请求从创建到结束的
change_request_response_time.json	PR请求从创建到首次响应的时长
change_requests_reviews.json	PR审阅者的数量
change_requests.json	PR的数量
code_change_lines_add.json	代码添加的行数
code_change_lines_remove.json	代码减少的行数
code_change_lines_sum.json	代码总变更数
contributor_email_suffixes.json	贡献者邮箱后缀
inactive_contributors.json	不活跃开发者数
issue_age.json	issue的生命时长（没有关闭的PR默认到23年3月份）
issue_comments.json	issue的评论数量
issue_resolution_duration.json	issue的开启到关闭的时长
issue_response_time.json	issue从开始到首次响应的时长
issues_and_change_request_active.json	issue和request的数量
issues_closed.json	issue关闭的数量
issues_new.json	issue创建的数量
new_contributors_detail.json	新增加的贡献者名单
new_contributors.json	新增加的贡献者数量



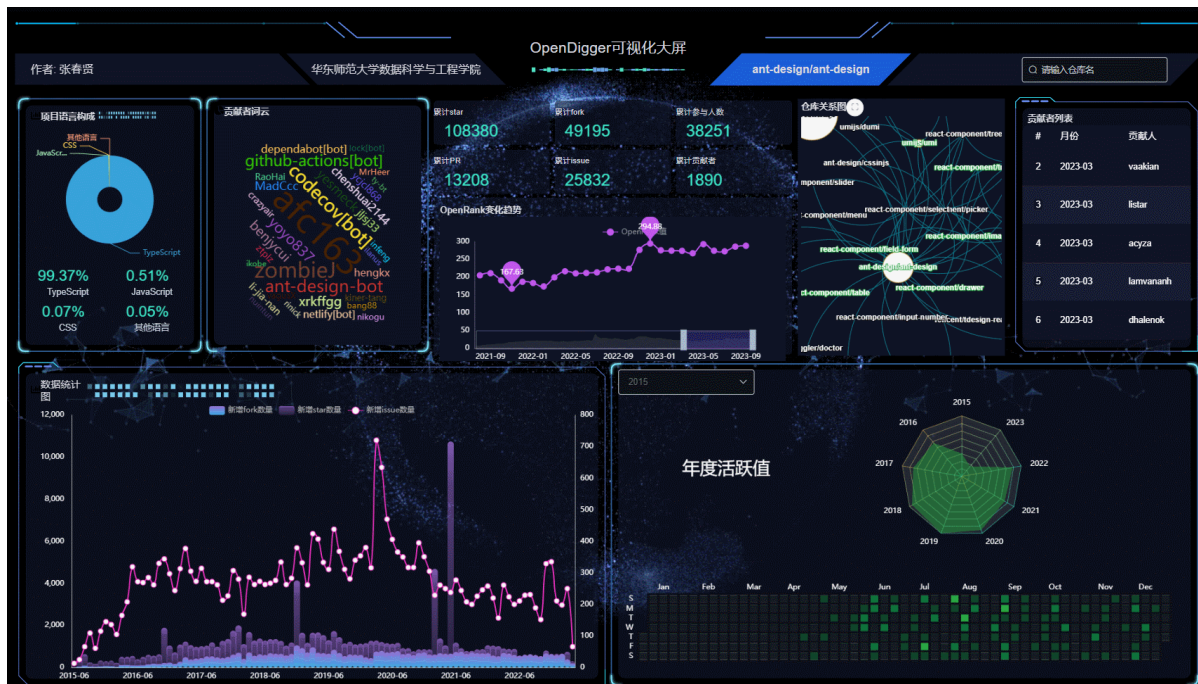
而Github API还可以查到一些额外的信息, 如项目语言分布图

openrank.json	openrank值
动态交互查询	
participants.json	项目参与者人数
start.json	start的数量
fork.json	fork数量

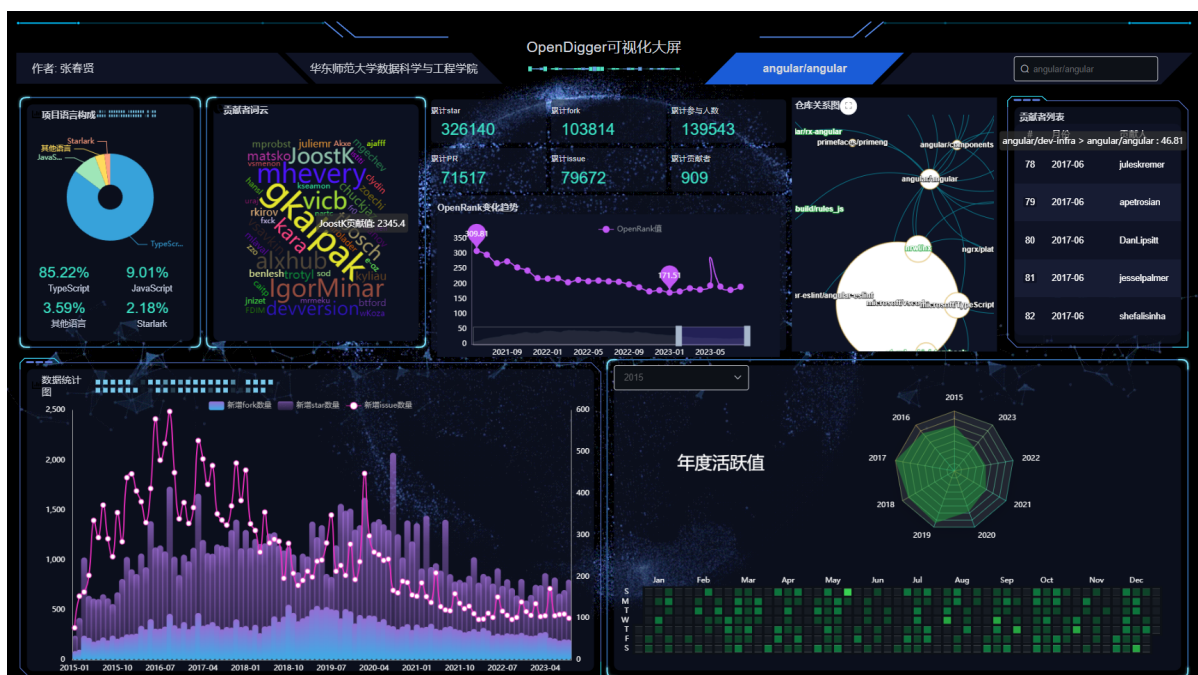
网站是动态可变, 有交互的!

不同于一般的可视化大屏, 本项目是可以进行查询的, 可以根据不同的开源项目展现不同的界面, 我们可以根据项目作者和项目名称对项目进行可视化查询.

查询的动作如下



输入查询后, 比如查询angular/angular项目, 就会从其他仓库的图表切换过来.



并且在每个项目下可以查询对应年份的活跃度

2016年apache/arrow活跃度



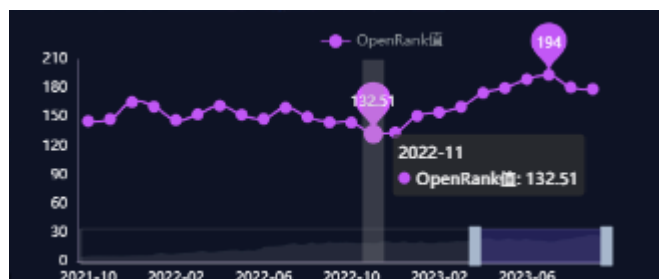
2017年apache/arrow活跃度



图表可交互

每一个图表, 在鼠标划过去的时候都是有运行的, 如波浪滚动, 列表滚动, 数值显示, 放大等.

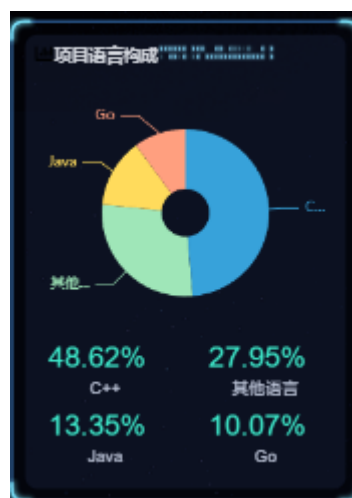




## 图表解释

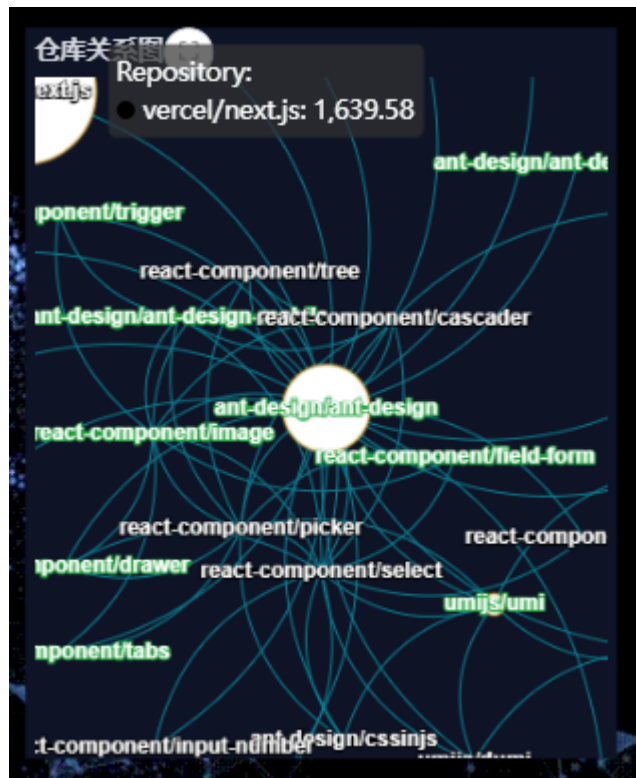
### 项目语言构成

通过Github API查询, 可以查到一个项目的语言构成情况, 对项目的技术栈有个基本的了解.



### 仓库网络

通过仓库网络可以查看当前仓库与其他仓库的联系, 体现github开源社区火热的合作氛围. 通过较大的点和连接值可以找到当前仓库的最相关仓库.



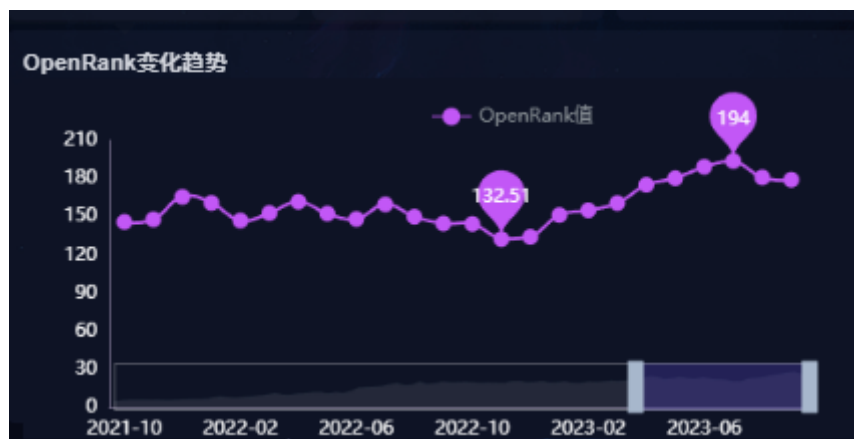
## 项目基本信息一览

查询项目的star, fork, 参与人数, issue, 贡献者数量, 看它就对了!



## OpenRank趋势

openRank是由OpenDigger独创的开源项目指标, 本大屏把它的变化趋势展示了出来



## 贡献者邮箱

通过贡献者邮箱列表, 可以了解到贡献者的一些国籍信息. 如gmail通常是欧美人, 163和qq通常是国人.





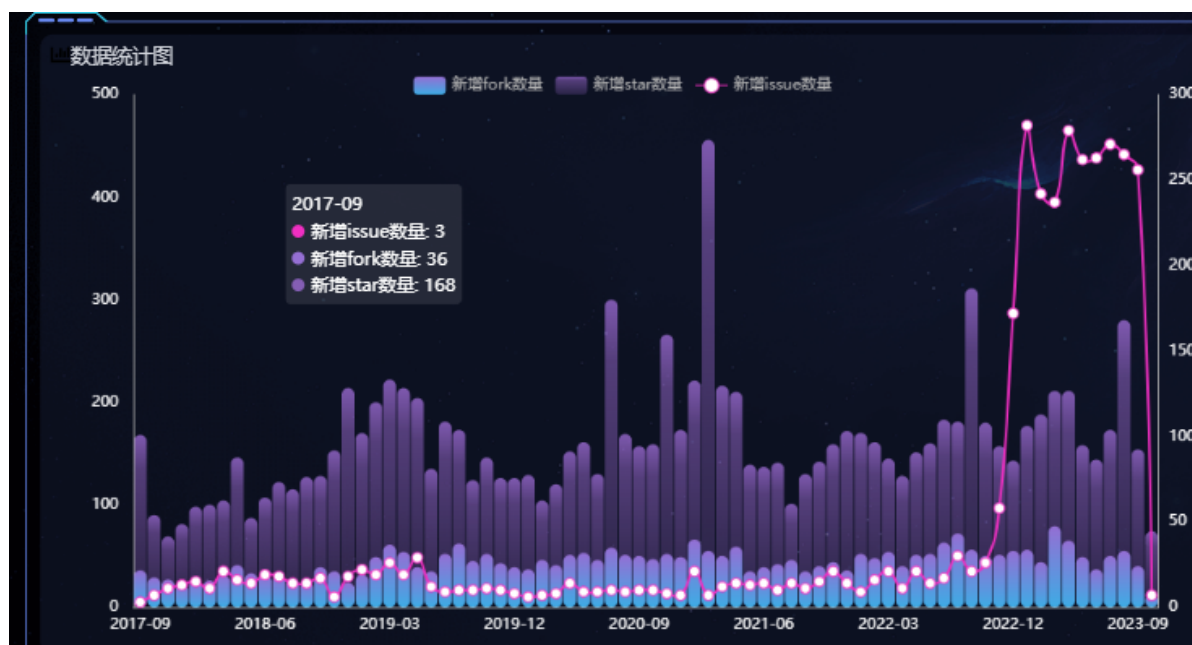
## 贡献者列表

可以查看贡献的大佬的名单! 而且是滚动名单

贡献者列表		
#	月份	贡献人
241	2022-07	zagto
242	2022-07	KBambrick1
243	2022-06	drin
244	2022-06	WilliamWhi...
245	2022-06	martin-liu

## 数据统计图

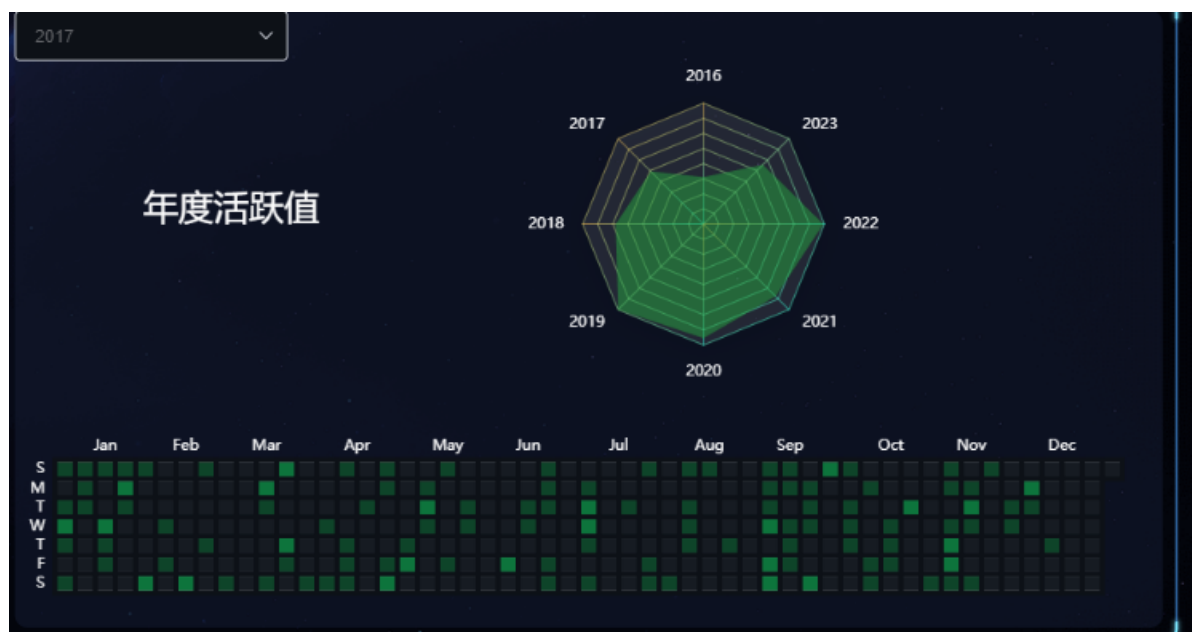
从star, fork和issue新增数量的角度, 查看一个开源项目的流行趋势, 三者越高越受欢迎!



## 年度活跃图

仿Github热力图, 开业查看项目的更新情况

雷达图可以查看项目的年度活跃情况, 让你对开源项目更好了解!



## 比赛收获

收获了一个完整的可视化项目经验, 从头到尾规划了如何更好的展示开源项目, 为开源助力