目录

[一、业务模型分析 1](#_Toc2882320)

[二、确定服务边界 2](#_Toc2882321)

[三、数据库的划分 3](#_Toc2882322)

[四、技术选型 4](#_Toc2882323)

[五、架构设计 5](#_Toc2882324)

[六、各服务接口设计 6](#_Toc2882325)

[七、主要功能活动图 7](#_Toc2882326)

[八、日志和监控 8](#_Toc2882327)

### 一、业务模型分析



房产的产权属于某个用户，并且需要经纪人进行代理；推荐需要对房产进行推荐（包括热门推荐和最新推荐），因此推荐依赖房产；经纪人属于某个经纪机构，同时经纪人也是某个用户，因此经纪人依赖于用户；评论可以针对百科和房产。因此，从图中可以看出，经纪人、经纪机构、用户这三个联系比较紧密，而推荐主要是针对房产，最后剩下了两个不太重要的评论和百科，通过以上分析，可以将整个项目分为三个服务，分别是用户服务、房产服务、评论服务。这样产生了的依赖关系是评论服务依赖用户服务，房产服务依赖用户服务和评论服务，用户服务作为底层服务、不依赖任何服务。另外还需要一个API GATEWAY服务，该服务主要进行鉴权、服务的聚合等功能，并且房产的评论正好通过该服务进行了聚合，即通过房产的id。最后形成的服务依赖之间的关系如下：



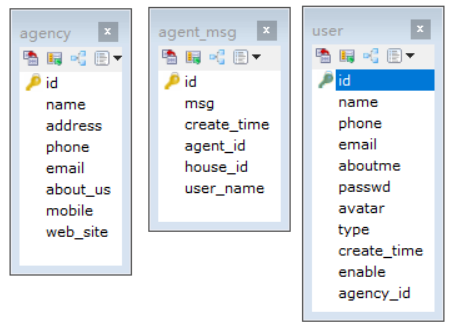
### 二、确定服务边界



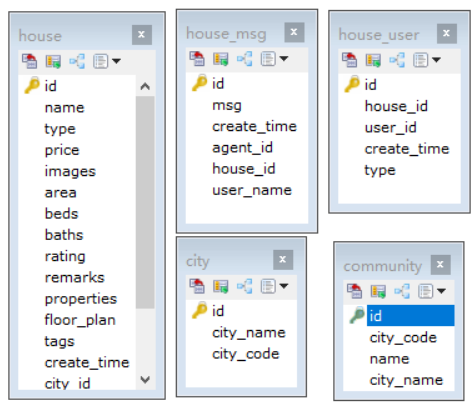
评论服务需要从用户服务中获取用户的姓名、头像等信息，不包括用户的密码、邮箱等信息；房产服务需要从用户服务中获取经纪人的头像、邮箱、姓名、手机等信息，需要知道经纪人的密码等信息。这样就确定了服务之间的边界，即评论服务和用户服务之间的边界是用户、用户服务和房产服务之家的边界是经纪人。

### 三、数据库的划分

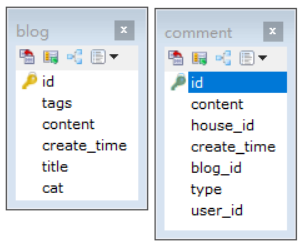
1、用户服务



2、房产服务



3、评论服务



### 四、技术选型



### 五、架构设计



### 六、各服务接口设计

1. 用户服务

1.1 注册(添加用户、激活用户)

1.2 鉴权（根据token查询用户）

1.3 登录/登出（身份认证（绑定token）、身份注销（解绑token））

1.4 经纪人/经纪机构（经纪机构列表/详情、添加经纪机构、经纪人列表/详情）

1.5 查询（根据重置key查询email、根据id查询、根据用户属性查用户列表）

1.6 更新（密码重置、密码重置邮件、用户属性更新）

2. 房产服务

2.1 查询（房产列表、房产详情）

2.2 添加（房屋添加、用户绑定）

2.3 推荐（热门房产、最新房产）

2.4 其它（评分）

3. 评论服务

3.1 查询（评论列表、评论详情）

3.2 添加（评论的添加）

4. API-Gateway接口设计

4.1 评论博客（查看评论、发布评论、查看博客、发布博客）

4.2 鉴权（触发动作、身份验证）

4.3 用户（注册激活、登录、用户修改、重置密码、登出、注册、经纪机构创建、经纪机构查询、经纪人查询）

4.4 房产（房产列表、房产新增、房产评分、房屋收藏/删除、房产详情）

4.5 首页

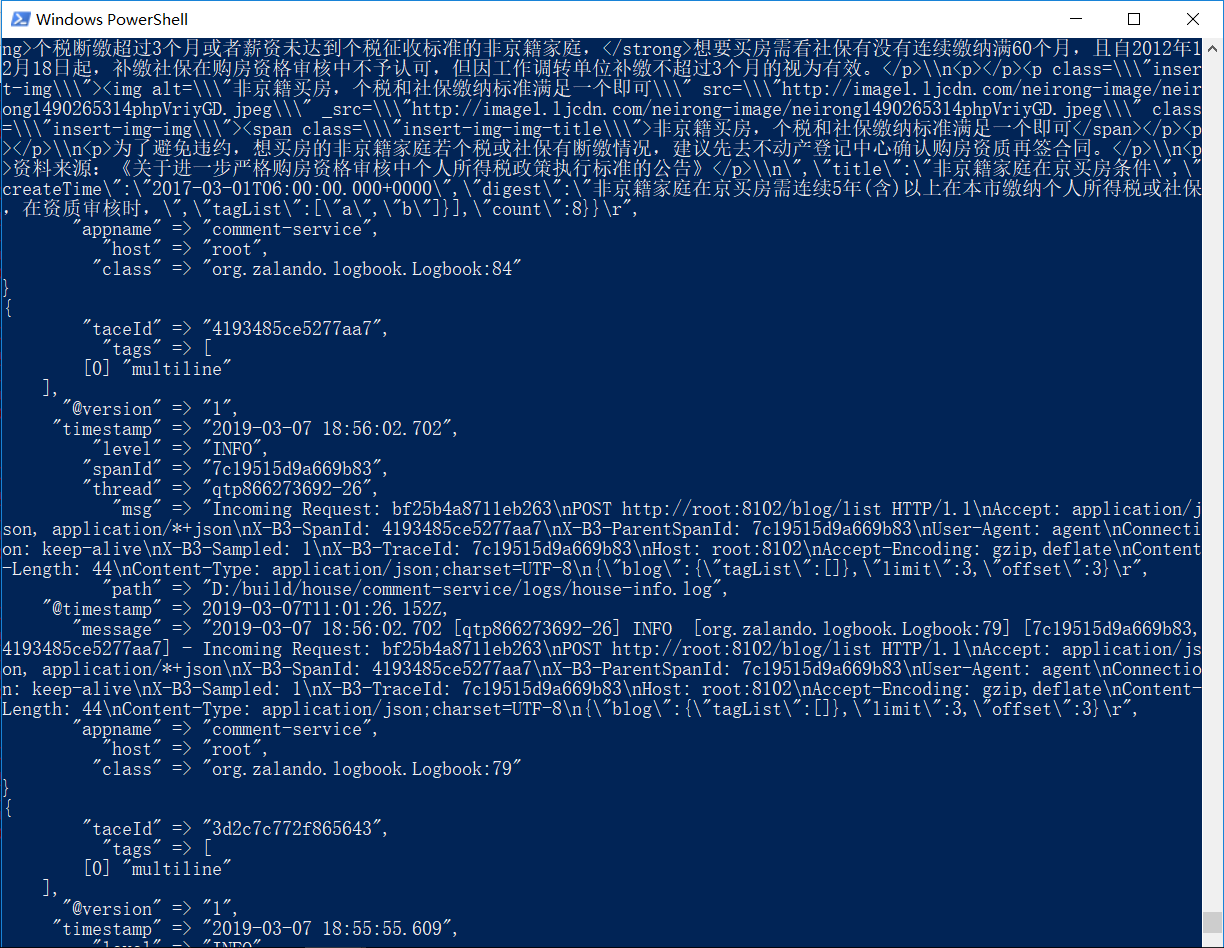
### 七、主要功能活动图

1、API-Gateway鉴权时序图

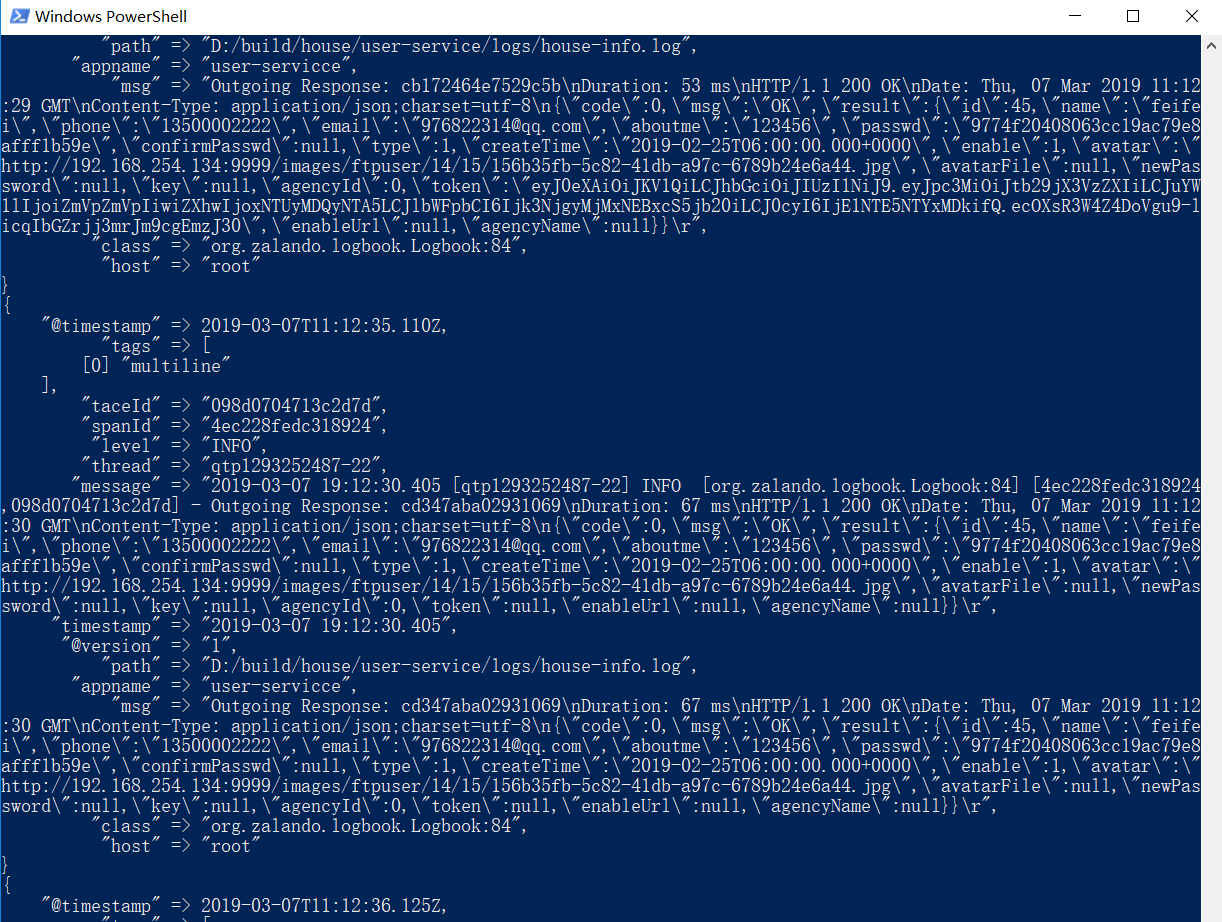


### 八、日志和监控

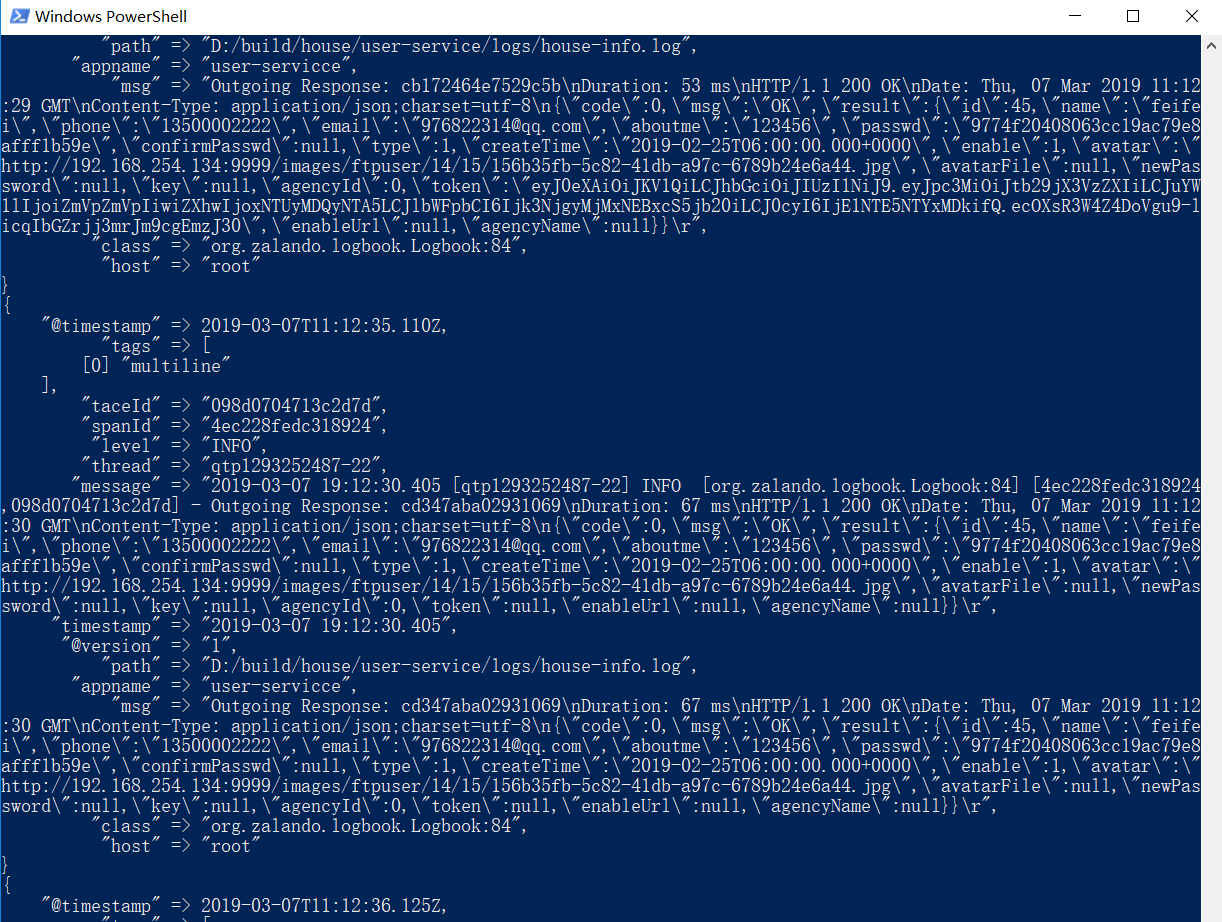
1、某段时间网关服务（api-gateway）、评论服务（comment-service）、房产服务（house-service）、用户服务（user-service）日志输出图如下：



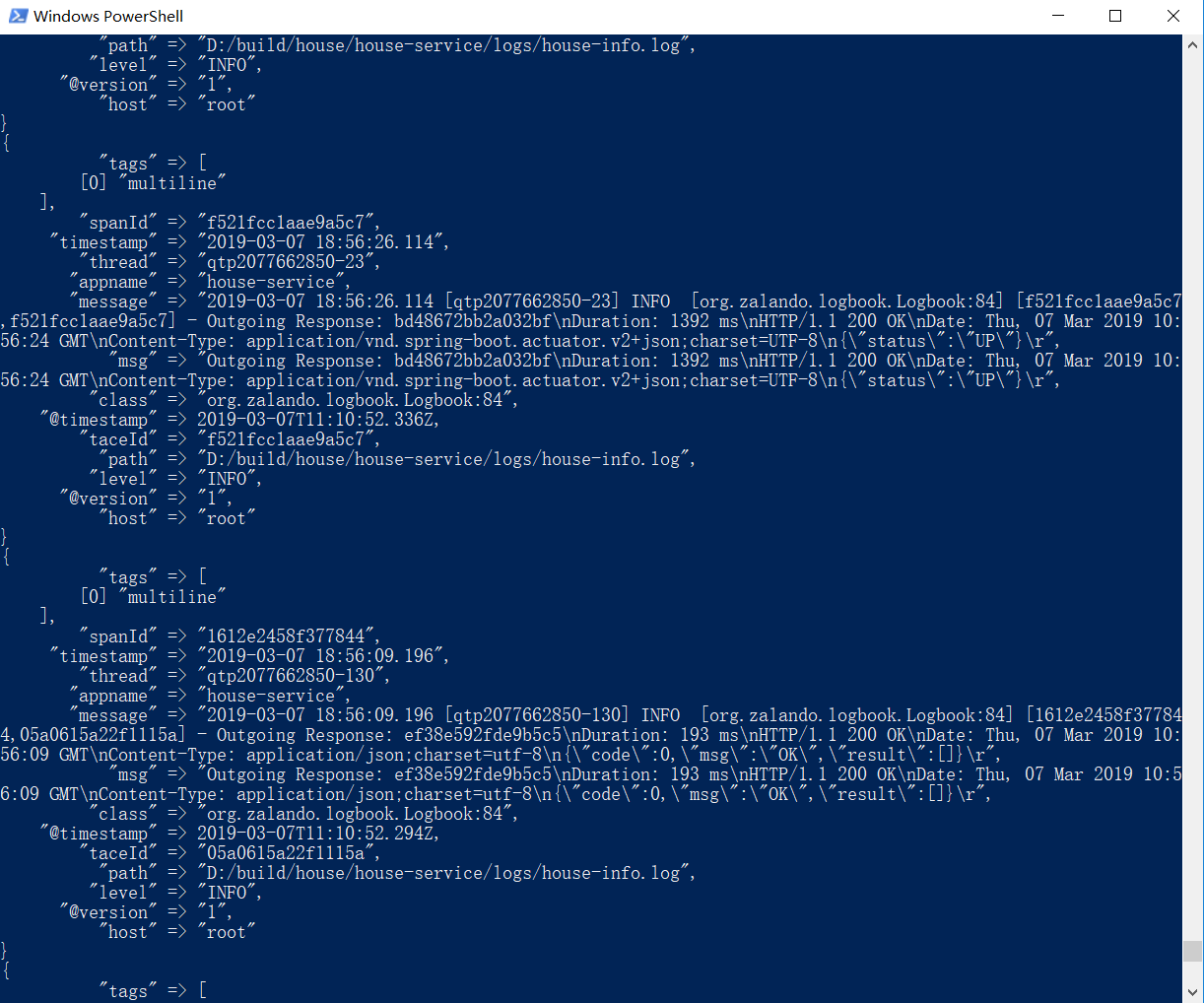
评论服务日志输出图



网关服务日志输出图

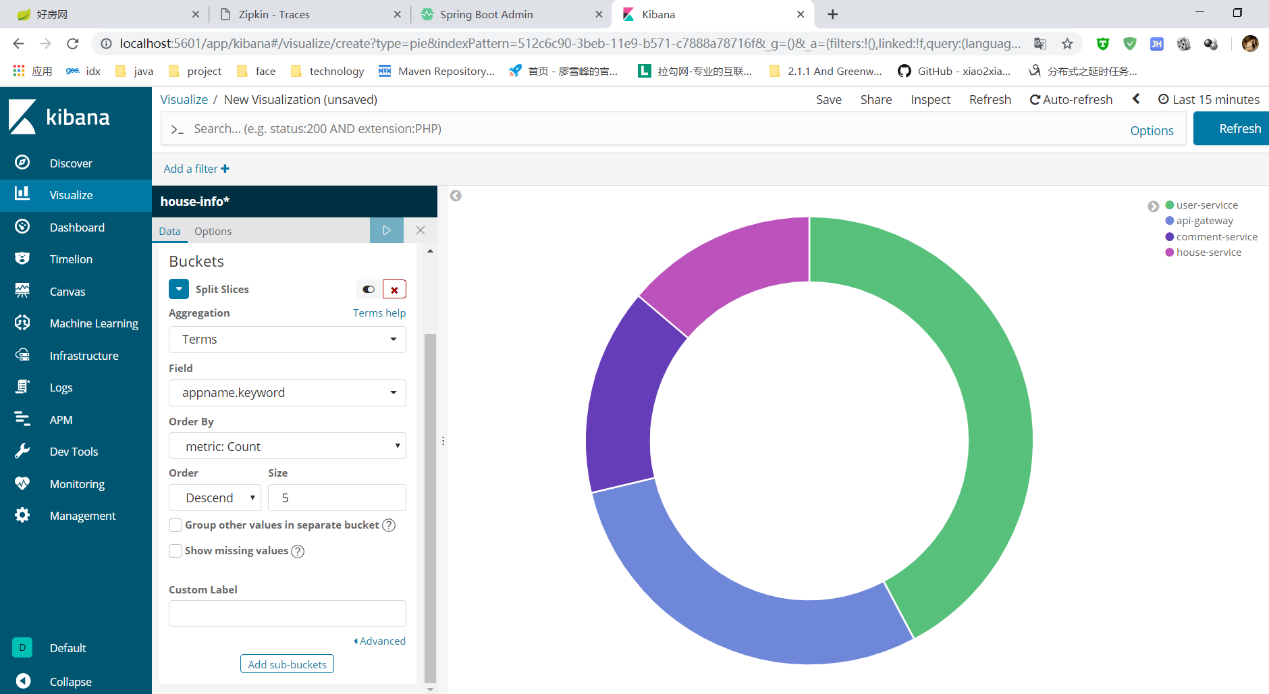


用户服务日志输出图



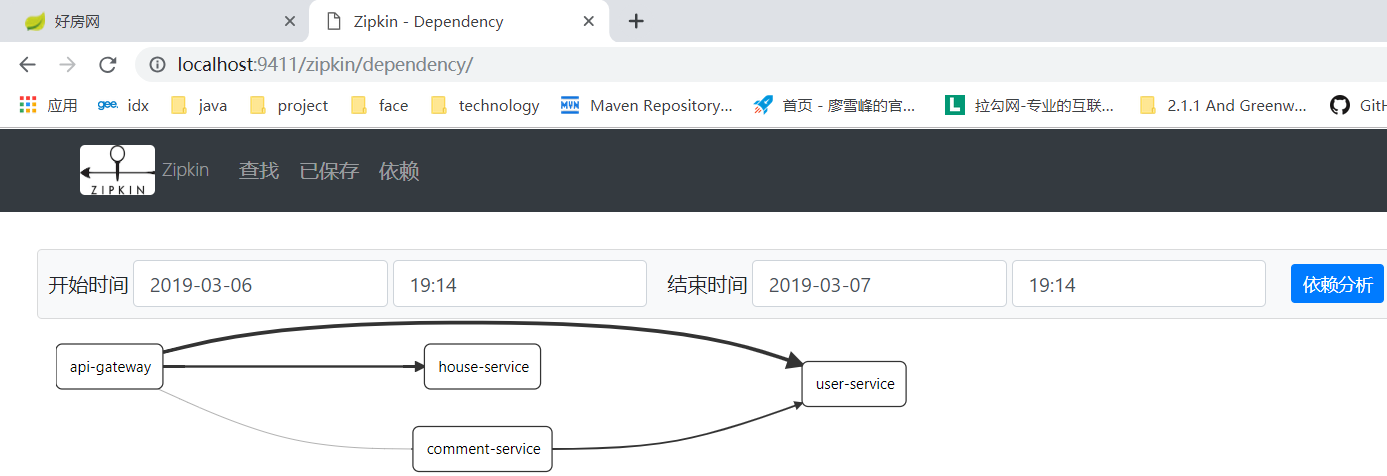
房产服务日志输出图

2、该段时间Kibana形成的日志输出饼图如下：



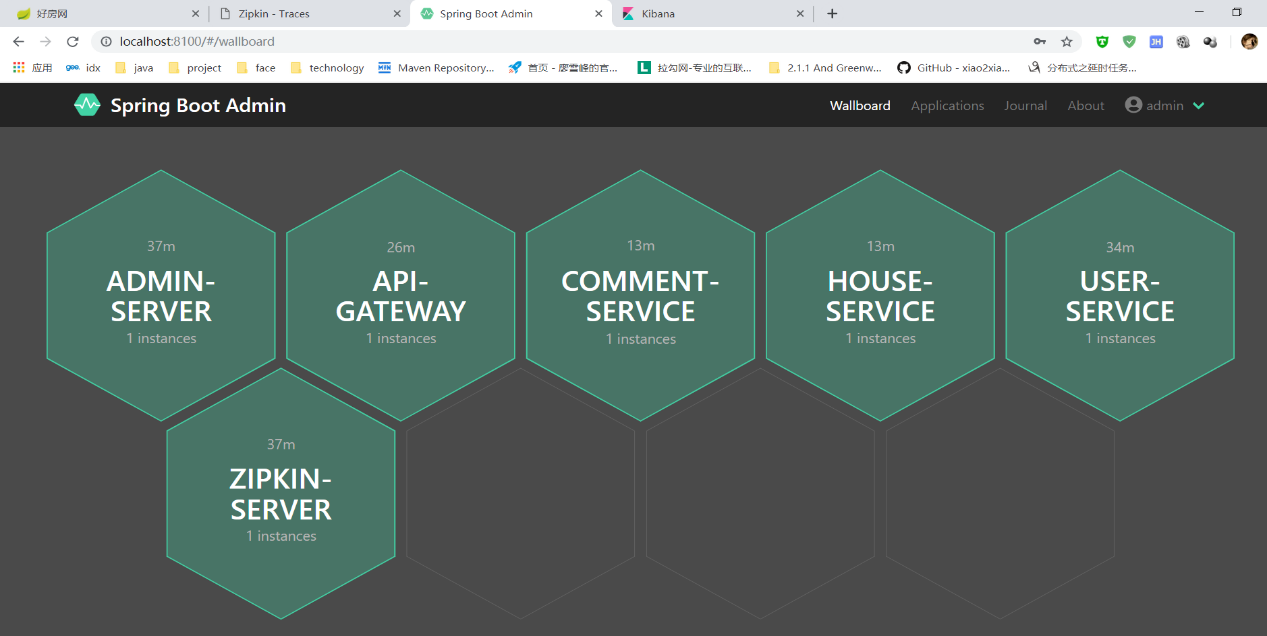
Kibana日志收集饼图

3、某段时间zipkin server链路追踪形成的关系依赖如下：

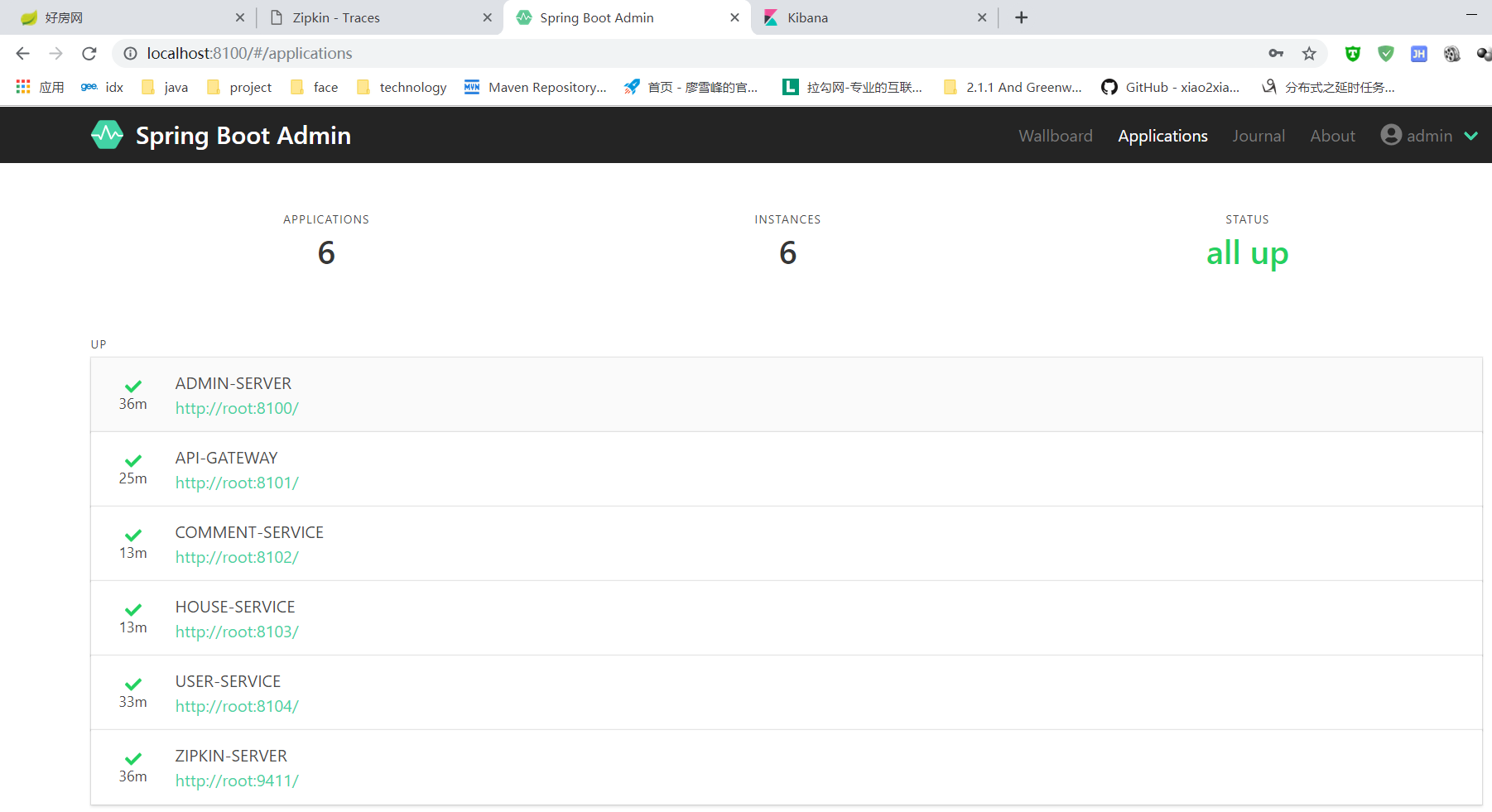


链路关系依赖

4、spring admin server监控服务的状态如下所示：



监控状态1



监控状态2