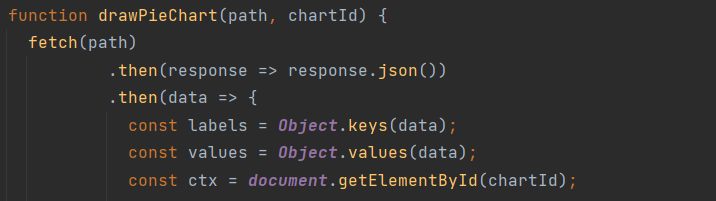
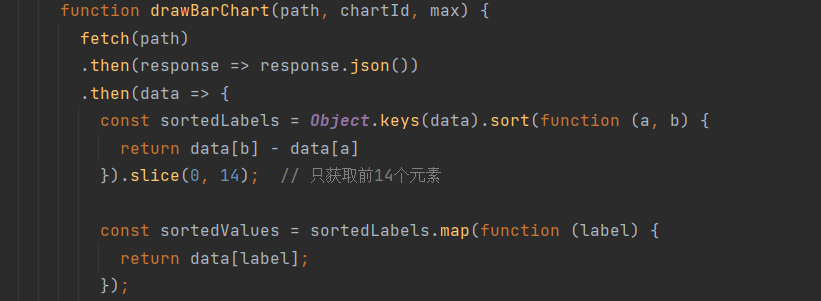
我们的可视化方式有三种：环状图、柱状图、单词云。每种可视化的方式都有相应的方法进行绘制。

环状图的方法如下，path为前端发送请求到达的路径，chartId表明要绘制的哪个数据图。

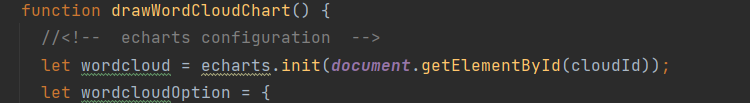


柱状图的方法如下，柱状图的排序方式均设置为从大到小，path为前端发送请求到达的路径，chartId表明要绘制的哪个数据图，max表示y轴最大值。



单词云的方法如下,path为前端发送请求到达的路径，cloudId表明要绘制的哪个数据图。

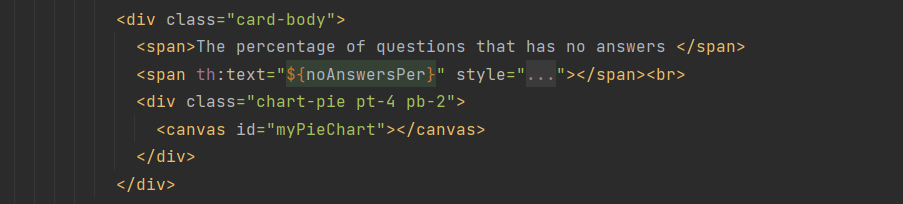




Numbers of Answers部分：

展示没有answers的百分比：

前端发送请求调用后端的代码或函数如下，第一个是百分比，第二个是环状图。

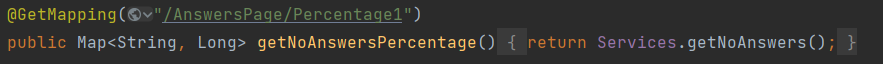




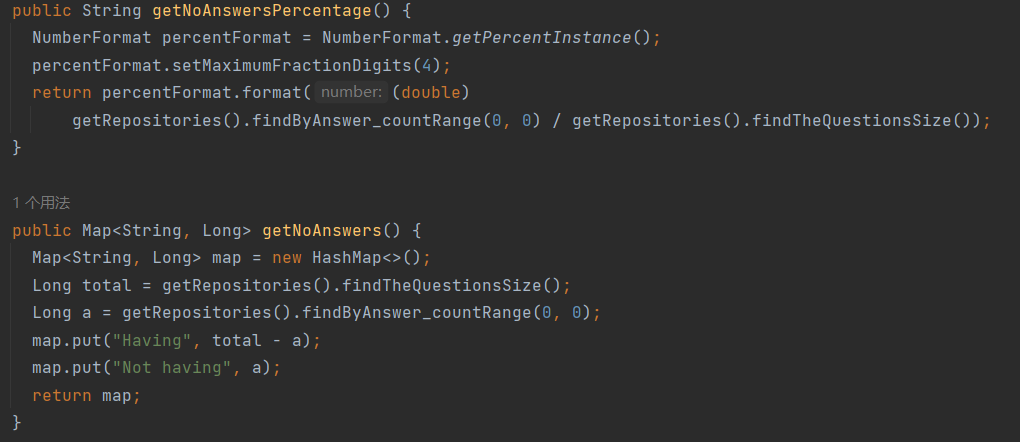
对应的ModelAndView方法如下



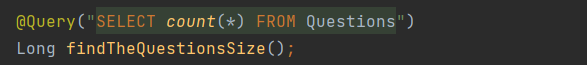
对应的RestController方法如下



相关的Services方法如下

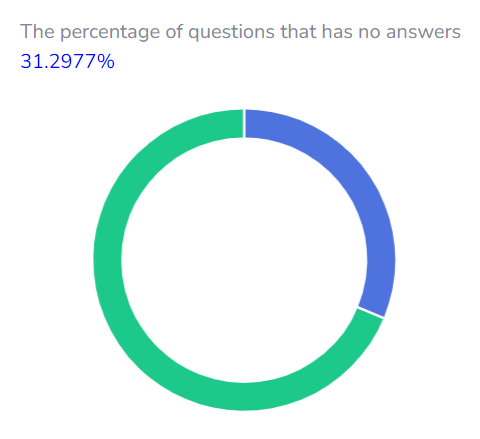


相关的Repositories数据库查询方法如下



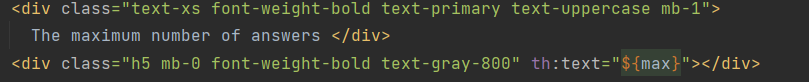


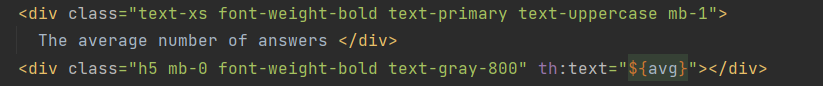
网页展现的结果如下



展示answers数量的最大值和平均值：

前端发送请求调用后端的代码或函数如下，第一个是最大值，第二个是平均值。

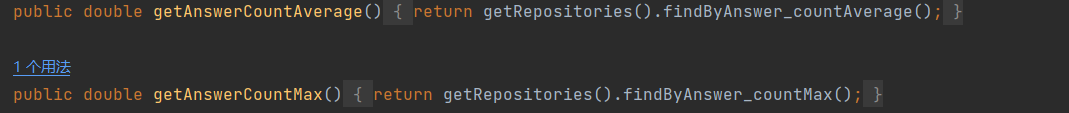




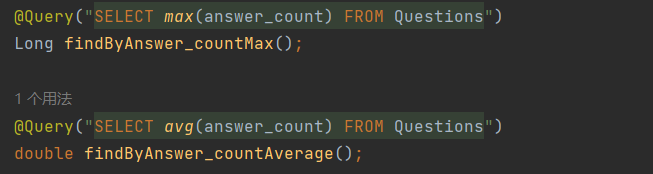
对应的ModelAndView方法如下



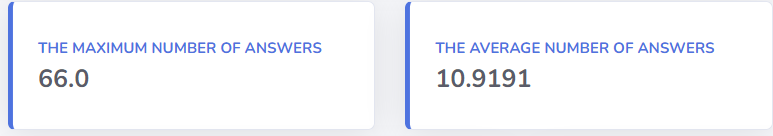
相关的Services方法如下



相关的Repositories数据库查询方法如下



网页展现的结果如下



展示answers数的分布：

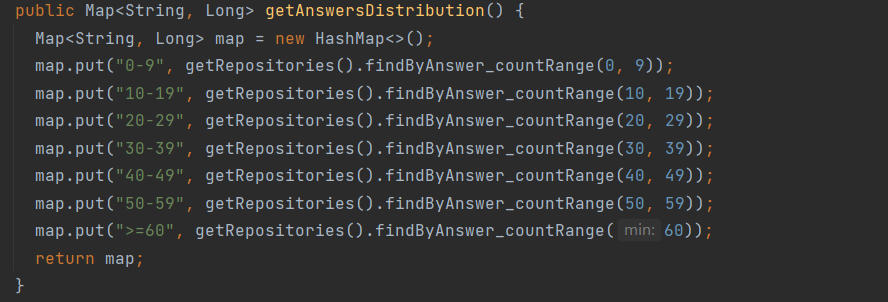
前端发送请求调用后端的代码或函数如下。



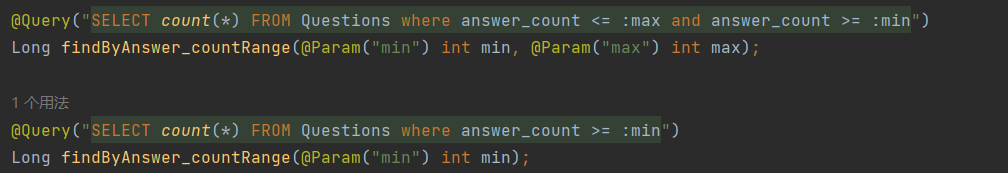
对应的RestController方法如下



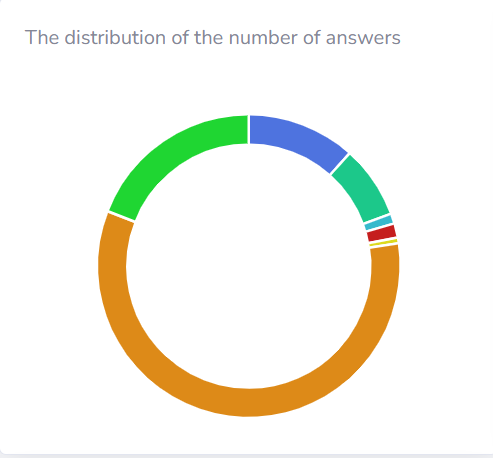
相关的Services方法如下，将Answer数划分为7个范围，统计每个范围的数量。



相关的Repositories数据库查询方法如下



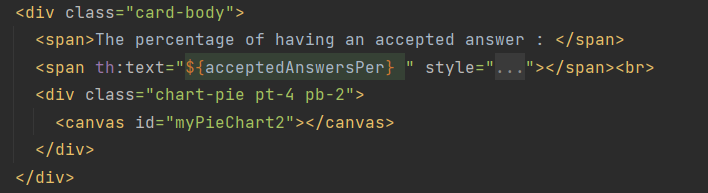
网页展现的结果如下



Accepted Answers部分：

展示有accepted answer的百分比：

前端发送请求调用后端的代码或函数如下，第一个是百分比，第二个是环状图。





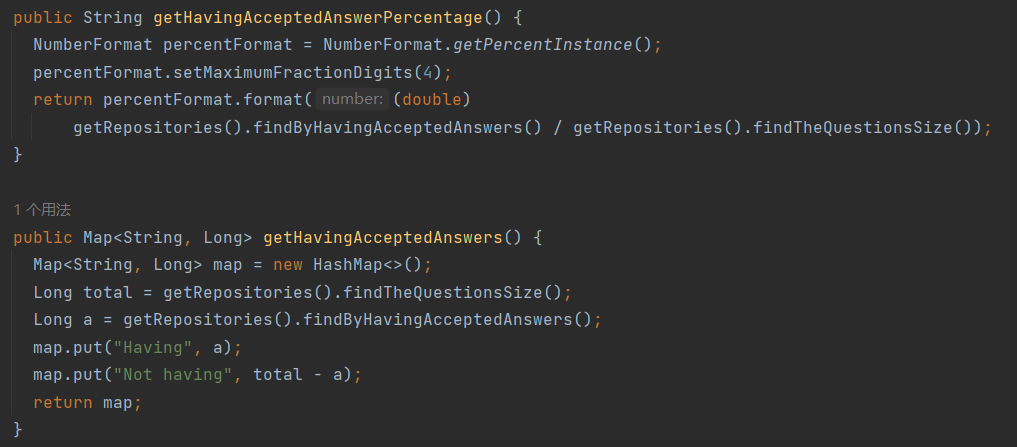
对应的ModelAndView方法如下



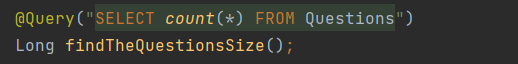
对应的RestController方法如下



相关的Services方法如下

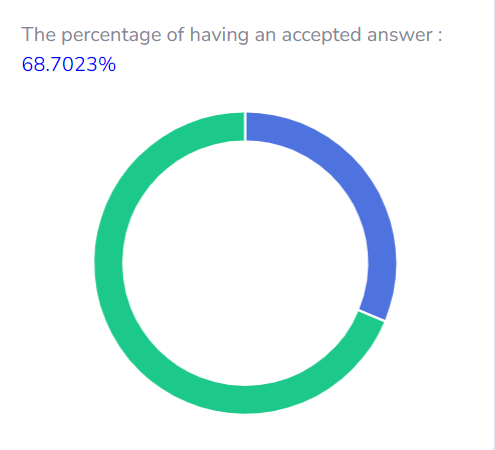


相关的Repositories数据库查询方法如下





网页展现的结果如下



展示问题从提出到解决时间间隔的分布：

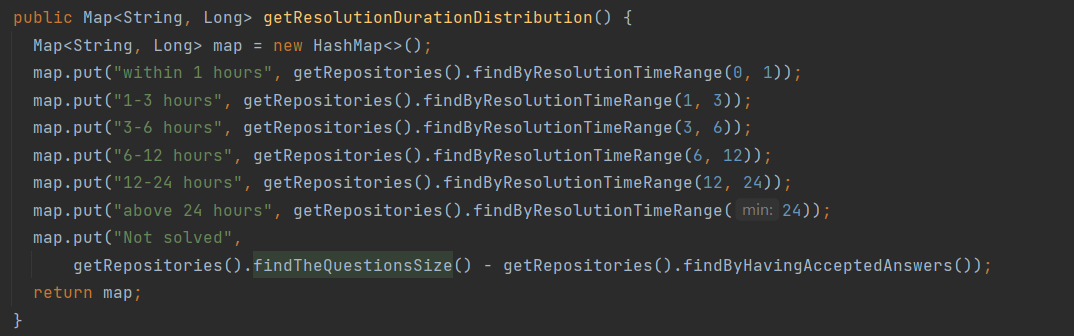
前端发送请求调用后端的代码或函数如下



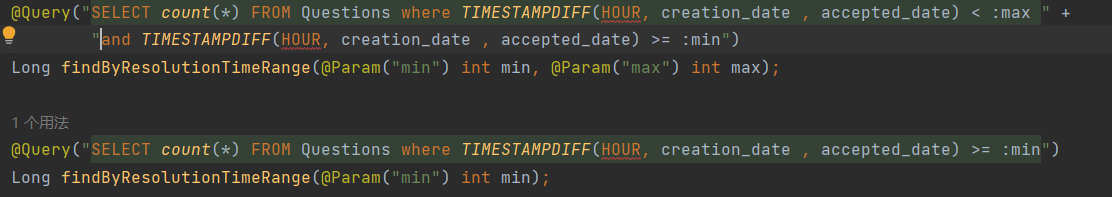
对应的RestController方法如下

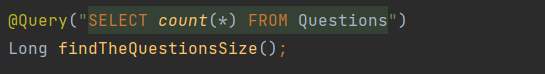


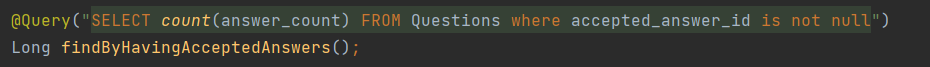
相关的Services方法如下，分为七个部分，有六个部分是按时间范围划分，还有一个部分是记录未解决问题的数量。



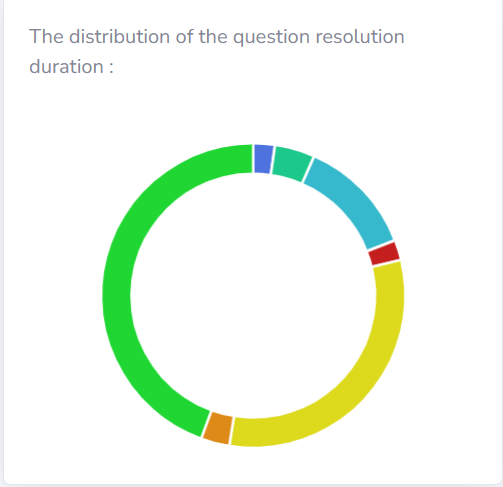
相关的Repositories数据库查询方法如下





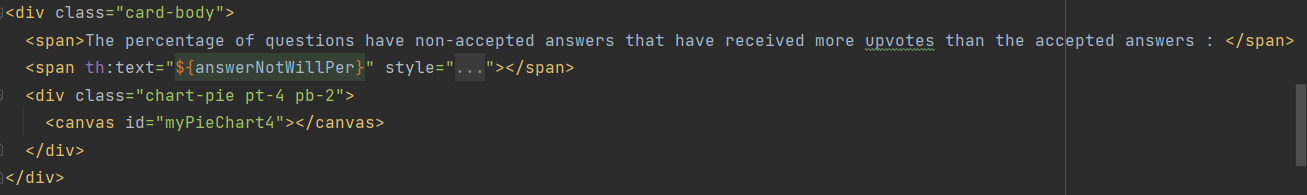


网页展现的结果如下



展示含有 non-accepted answer 的 upvote 数高于 accepted answer 的问题的百分比：

前端发送请求调用后端的代码或函数如下，第一个是百分比，第二个是环状图。

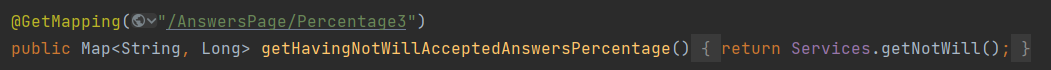




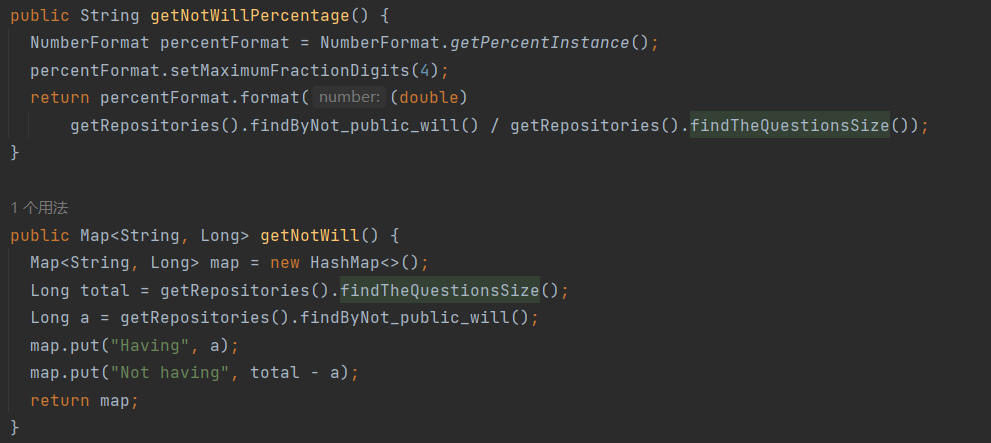
对应的ModelAndView方法如下



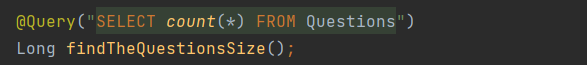
对应的RestController方法如下

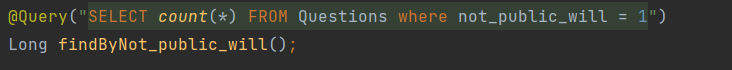


相关的Services方法如下

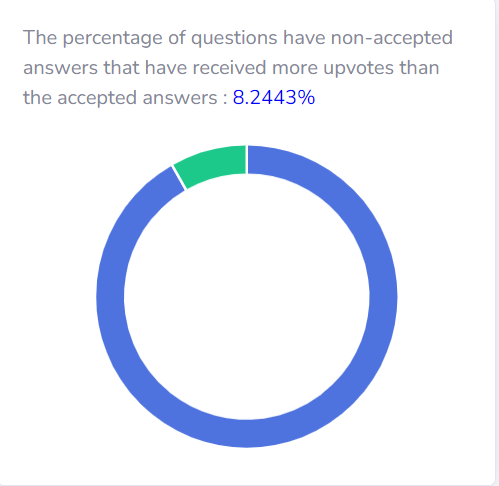


相关的Repositories数据库查询方法如下





网页展现的结果如下



Tags部分：

展示哪些 tags 经常和 Java tag 一起出现 ：

前端发送请求调用后端的代码或函数如下，第一个是tag，第二个是柱状图,第三个是单词云。







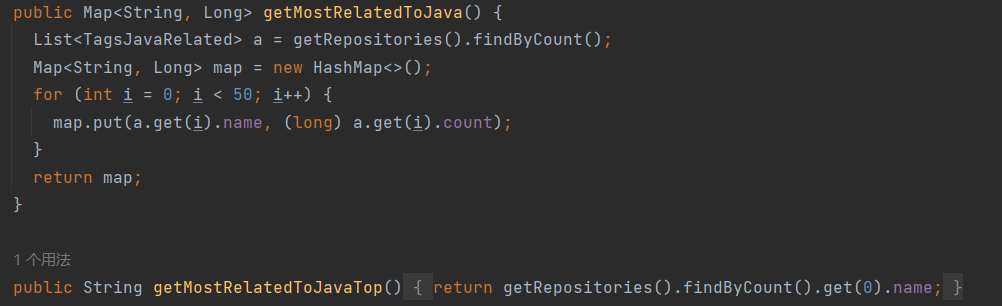
对应的ModelAndView方法如下



对应的RestController方法如下



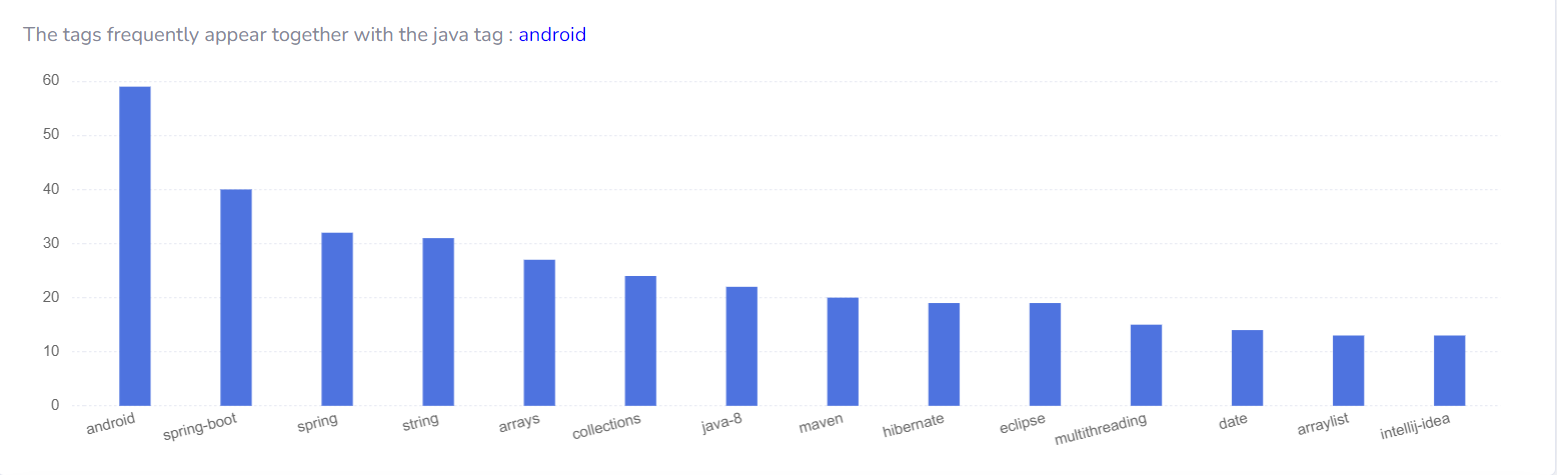
相关的Services方法如下，按降序获取了前50条数据。



相关的Repositories数据库查询方法如下



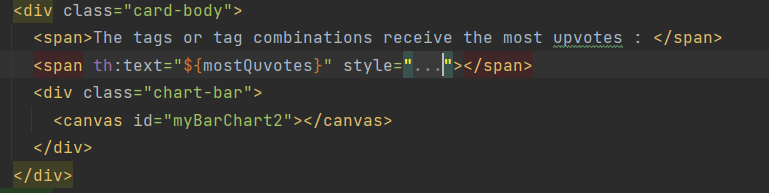
网页展现的结果如下





展示哪些 tags 或 tag 的组合得到最多的 upvotes ：

前端发送请求调用后端的代码或函数如下，第一个是tag 或 tags 的组合，第二个是柱状图。

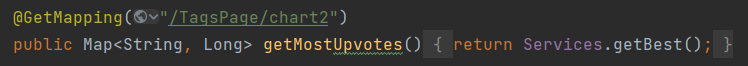




对应的ModelAndView方法如下



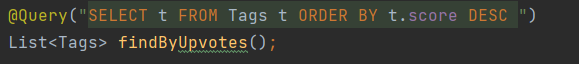
对应的RestController方法如下



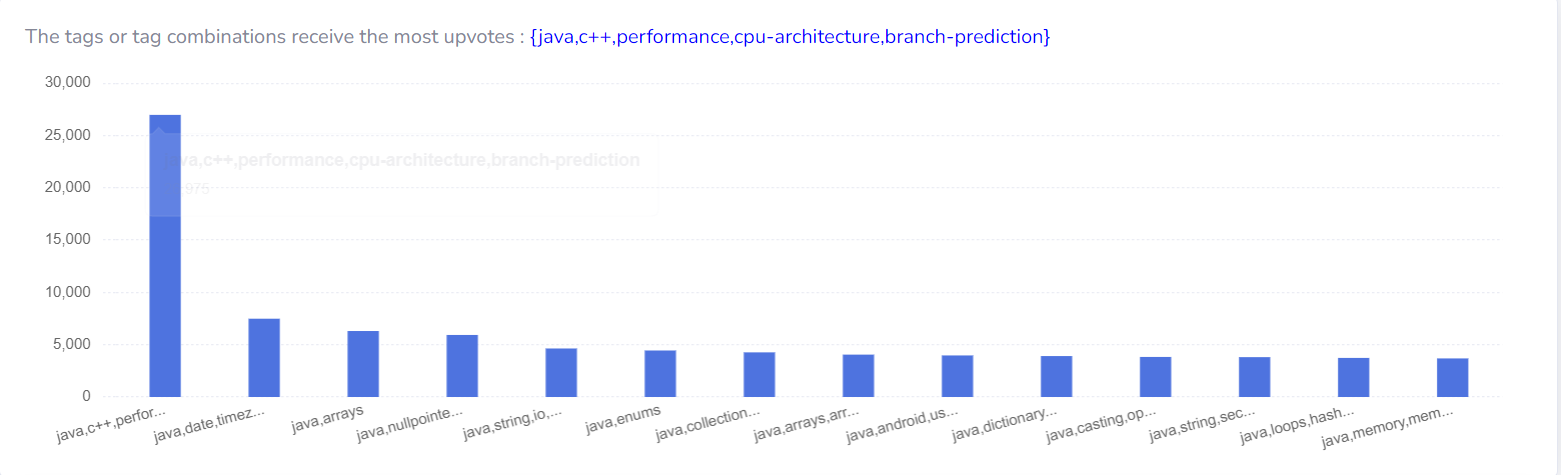
相关的Services方法如下，按降序获取了前14条数据。



相关的Repositories数据库查询方法如下

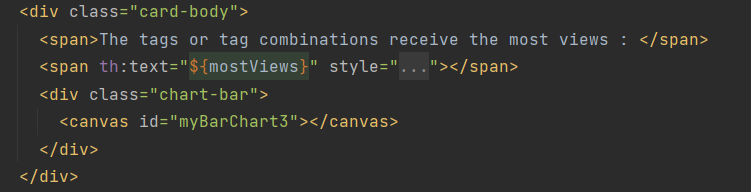


网页展现的结果如下



展示哪些 tags 或 tag 的组合得到最多的 views ：

前端发送请求调用后端的代码或函数如下，第一个是tag 或 tags 的组合，第二个是柱状图。





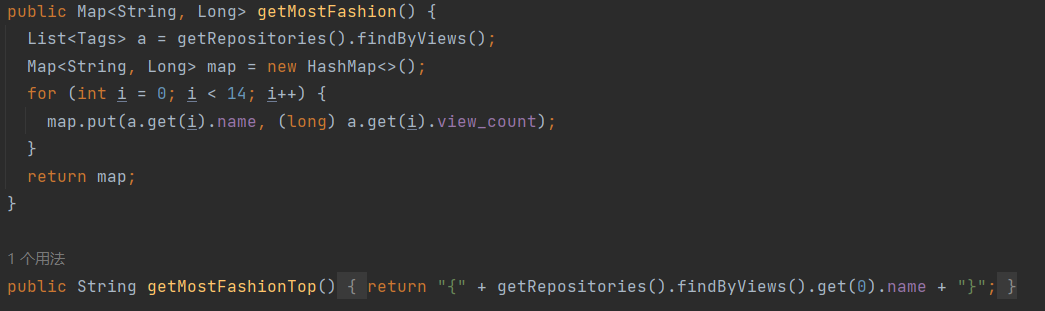
对应的ModelAndView方法如下



对应的RestController方法如下



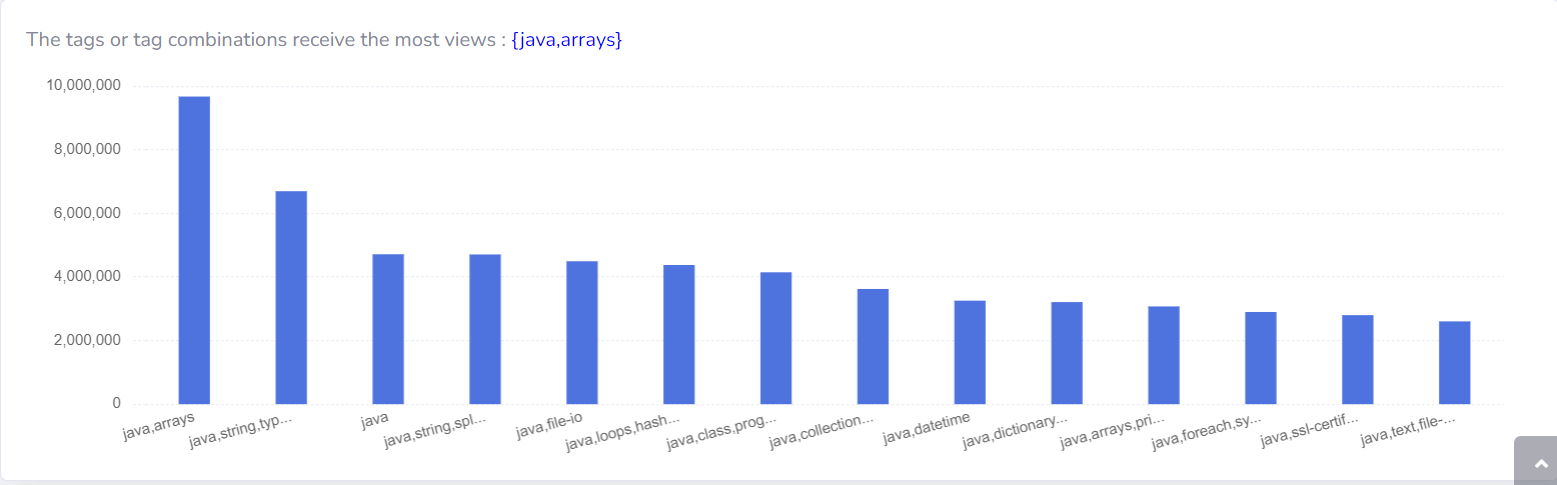
相关的Services方法如下，按降序获取前14条数据。



相关的Repositories数据库查询方法如下



网页展现的结果如下



Users部分：

展示参与 Thread 讨论的用户数量的分布 ：

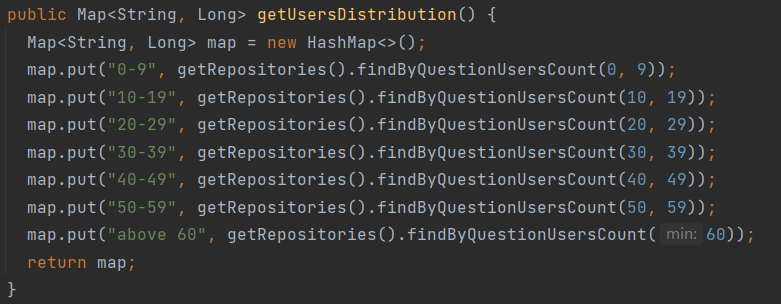
前端发送请求调用后端的代码或函数如下



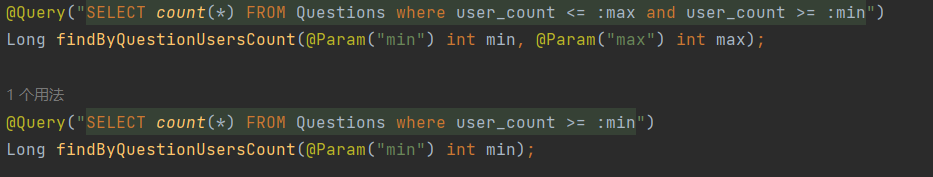
对应的RestController方法如下



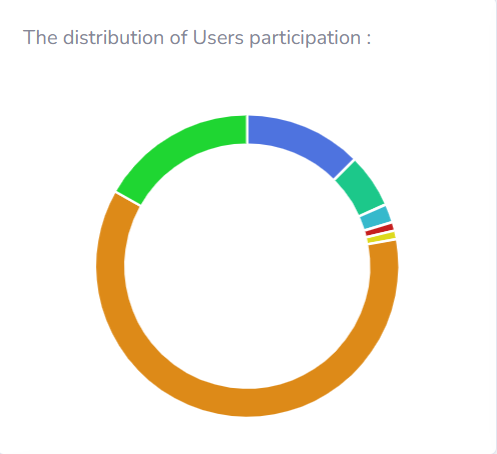
相关的Services方法如下，按照用户数量（在一个Thread里相同的用户只记录一次）分为7个范围。



相关的Repositories数据库查询方法如下



网页展现的结果如下

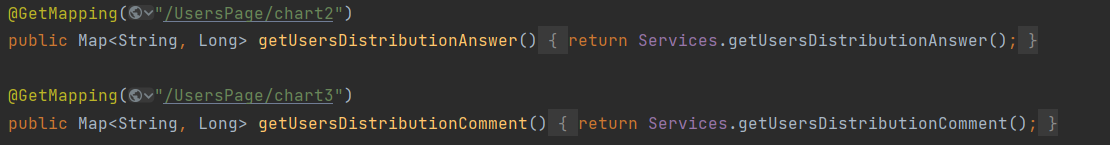


从问题回答者和评论者两个角度进行统计 ：

前端发送请求调用后端的代码或函数如下，第一个是answer方面的环状图，第二个是comment方面的环状图。



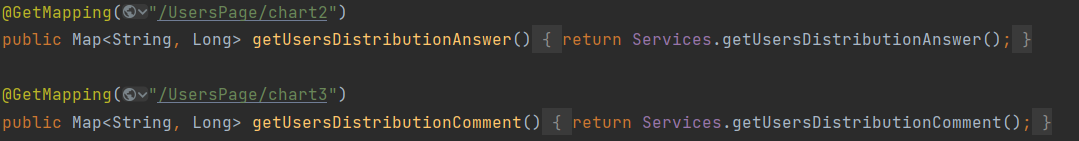
对应的RestController方法如下



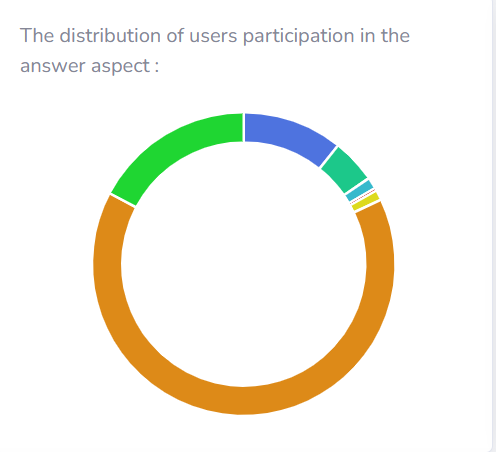
相关的Services方法如下，按照用户数量（在一个Thread里相同的用户只记录一次），在answer方面分为7个范围，在comment方面分为4个范围。

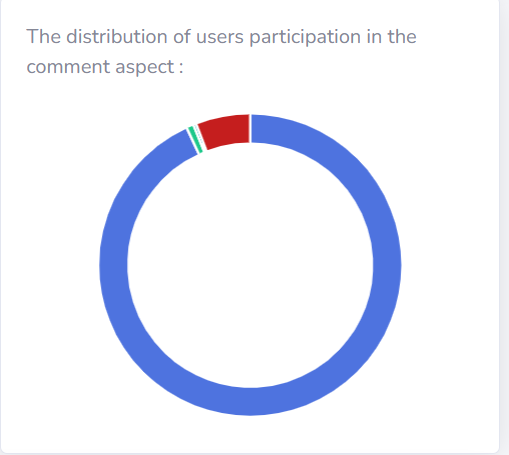


相关的Repositories数据库查询方法如下



网页展现的结果如下





展示哪些用户参与 java thread 讨论最活跃 ：

前端发送请求调用后端的代码或函数如下，第一个是用户，第二个是柱状图。

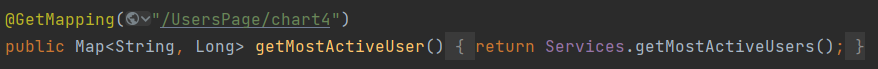




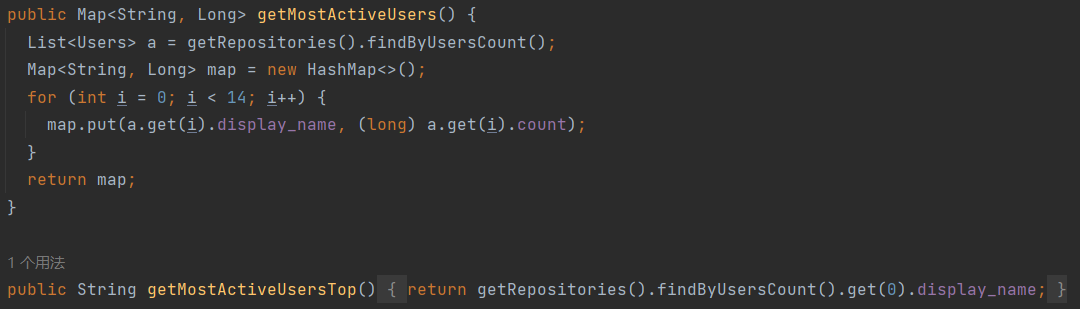
对应的ModelAndView方法如下



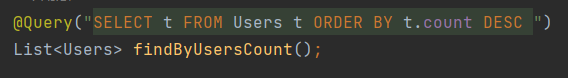
对应的RestController方法如下



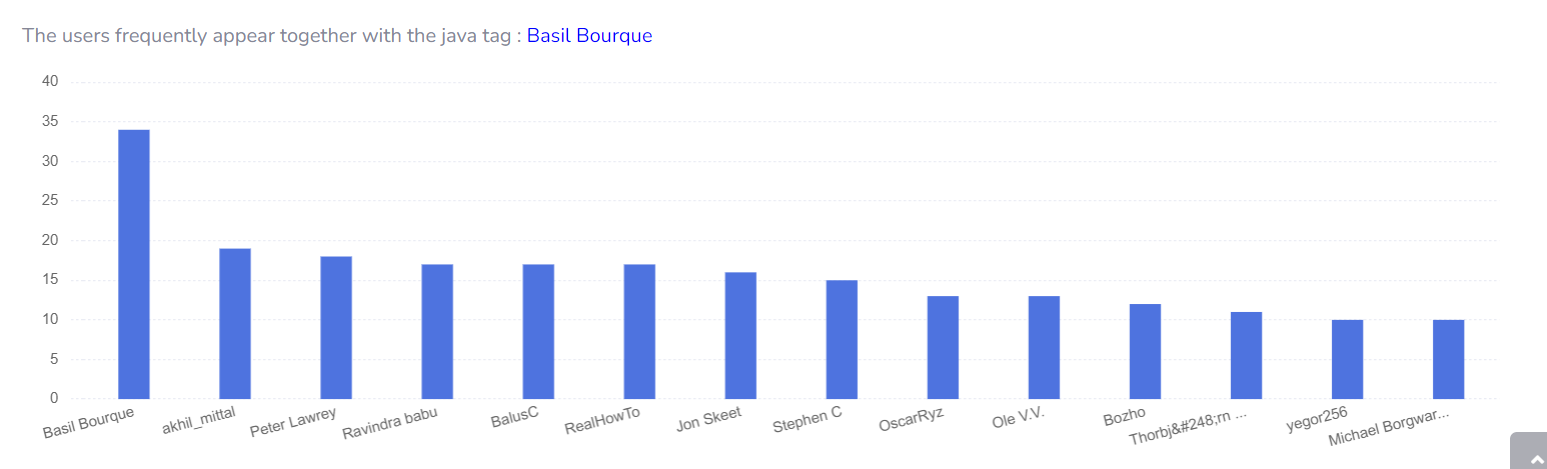
相关的Services方法如下，按降序获取前14条数据。



相关的Repositories数据库查询方法如下



网页展现的结果如下



Frequently discuss Java Apis部分：

前端发送请求调用后端的代码或函数如下，第一个是Api，第二个是柱状图，第三个是单词云。



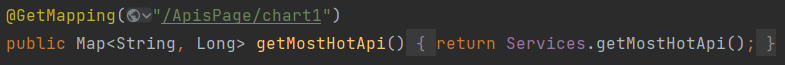




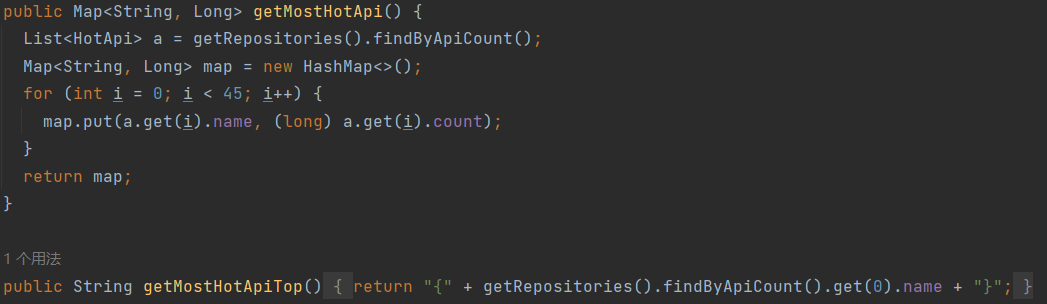
对应的ModelAndView方法如下



对应的RestController方法如下



相关的Services方法如下，按降序获取前45条数据。



相关的Repositories数据库查询方法如下



展示哪些 Java APIs 在 Stack Overflow 上最频繁被讨论

