豆瓣一刻项目思路：

1. 分析。

有不变的页面，也有变化的页面，这个项目中是侧边栏和主页的搜索部分，因此按每个页面中都有这两个不动的部分。这个单页面应用，根据锚点的变化请求相应的数据。

需要用到的技术：单页面应用，不同的控制器对应不同的视图。用less编辑css。整个布局采用rem匹配各种屏幕。请求豆瓣的数据用抓包工具，通过手机连接电脑（前提是手机电脑必须是同一个网络），让电脑成为代理服务器，手机访问豆瓣，抓包工具拦截数据，返回给后台。后台对数据进行处理，返还给浏览器，浏览器的$http获取数据后把内容展示到对应的html界面上。完成所有的页面后再用gulp项目构建工具，下载各中需要的插件，对代码进行压缩，合并，less预处理等构建任务，最后push到github上

1. 归类。

Views文件夹放各种页面的视图

Public文件夹 里面放less文件，images图片，图标字体fonts，还有libs文件夹，里面放后面在项目构建的时候用到的引用js文件。

Uploads中放加载的图片，在本项目是作者头像

Api文件夹 里面放各种页面的后台接口，也就是php文件

Script文件夹 放一些执行逻辑的js代码，本项目中用到的模块app的js代码，各种控制器集成的js，指令direction集成的js，服务集成的js

Index.html 主页面

.ico 页面图标图片

1. 写每个页面的html代码。用less编辑css。
2. 在根目录中初始化git仓库，创建分支developer，在这个分支上创建framework，这个分支上开发每个功能后提交到release分支，检查无误后提交到master分支，否则返回developer分支。等所有的功能开发好了后push到github仓库上（最后一步）
3. 开始引用angular.js的知识开发。主页index.html引用angular.js和angular-route.js模板还有scripts中的js文件。
4. App.js中创建模块，这个模块依赖路由模块和控制器模块。控制器大模块掌管着today，older等的控制器。这里面主要是对路由进行配置，不同的锚点对应不同的模板。以及对主页左上角的按钮进行来回的切换功能。
5. 控制器的js文件中是每个控制器下，依赖模块$http访问后台php后台请求豆瓣数据返回给浏览器，浏览器把对应的数据展示到不同的视图中
6. 在用node.js的npm下载gulp进行项目构建，下载各种构建用的插件，新建任务清单文件gulpfile.js文件，创建任务，找出路径，通过gulp处理后保存，最后执行命令。验证是否构建成功
7. 最后上传到github上