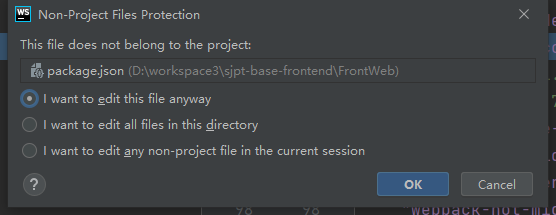
Webstorm 功能清单

流程图

Webstorm的增删改查的功能清单

[IMG_256](https://www.cnblogs.com/ElvisZhongShao/)

Webstorm配置化解析

WebstormUI

Webstorm内存

工作146:webstorm持续updating indices解决方法

歌谣--前端学习--前端日常工作

专栏收录该内容

334 篇文章5 订阅

订阅专栏

现象

在应用Webstorm进行开发相关工作时，发现IDE一直处于updating indices状态，根据错误提示，查看项目目录中.idea\workspace.xml文件

，发现在进行git版本管理过程中，此文件进行了merge操作，结果导致文件中内容格式错乱，通过回退版本，发现问题并为解决，通过网络查询，

可能是由于项目中node\_modules文件文件过大，而且将该文件夹作为目录标记，解决方法就是右键node\_modules，

**选择Mark Directory as->Excluded,会使IDEA排除index这个文件夹，问题得以解决。**

————————————————

版权声明：本文为CSDN博主「普通网友」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin\_43392489/article/details/110069425

**[添加文件夹右键以webstrom打开](https://www.cnblogs.com/ElvisZhongShao/p/15174499.html)**

安装引导过程有一项，可以选择直接添加。如果当时没有勾选，则需要自行添加注册表。

[原文地址](https://blog.csdn.net/qq_37086982/article/details/107366333" \o "原文地址：" \t "https://www.cnblogs.com/ElvisZhongShao/p/_blank)：[https://blog.csdn.net/qq\_37086982/article/details/107366333](https://blog.csdn.net/qq_37086982/article/details/107366333" \t "https://www.cnblogs.com/ElvisZhongShao/p/_blank)

win+ r 输入regedit .打开注册表

找到HKEY\_CLASSES\_ROOT \ Directory \ shell

在shell内新建项,命名为WebStorm

在WebStorm项上右键新建字符串值 icon

WebStorm项内右侧双击默认数值数据内填写你右键菜单的名称WebStorm打开此文件夹

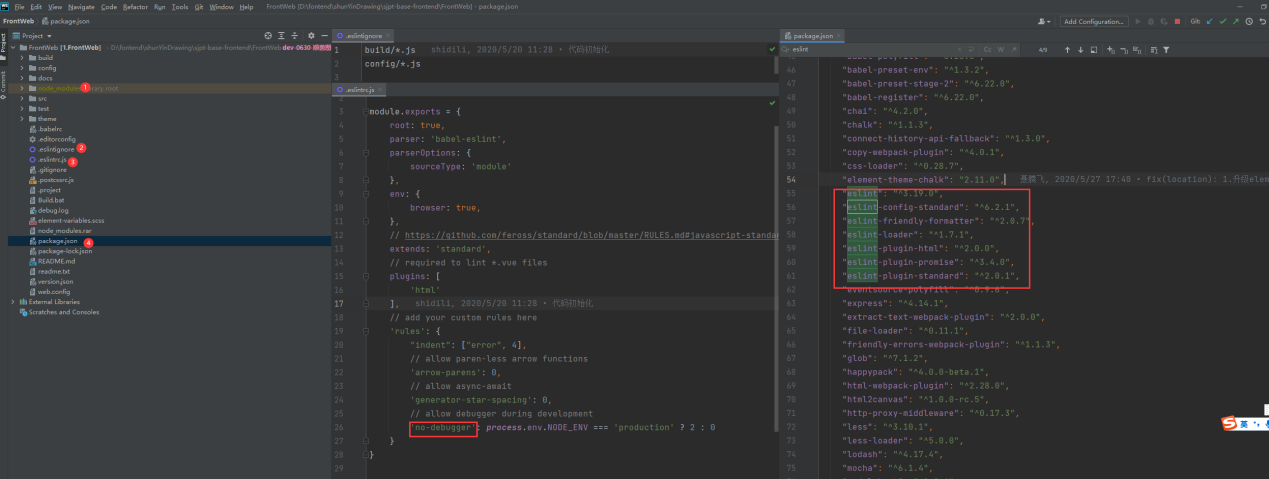
双击icon数值数据内填写  
C:\Program Files\JetBrains\WebStorm 2020.1\bin\webstorm64.exe

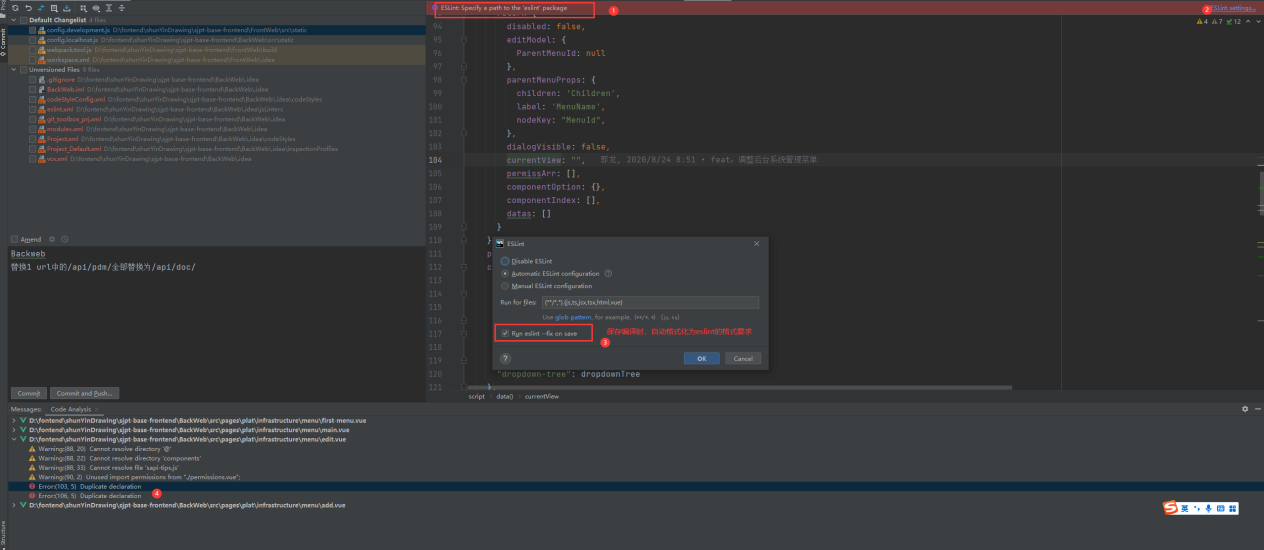
WebStorm项下新建command项  
command项内默认填写  
C:\Program Files\JetBrains\WebStorm 2020.1\bin\webstorm64.exe %1  
路径末尾添加%1(%前有一个英文空格)

英文单词

External change外部改变







# <https://juejin.cn/post/6963835326821302303>

**webstorm自动保存功能设置**

工作问题小结：

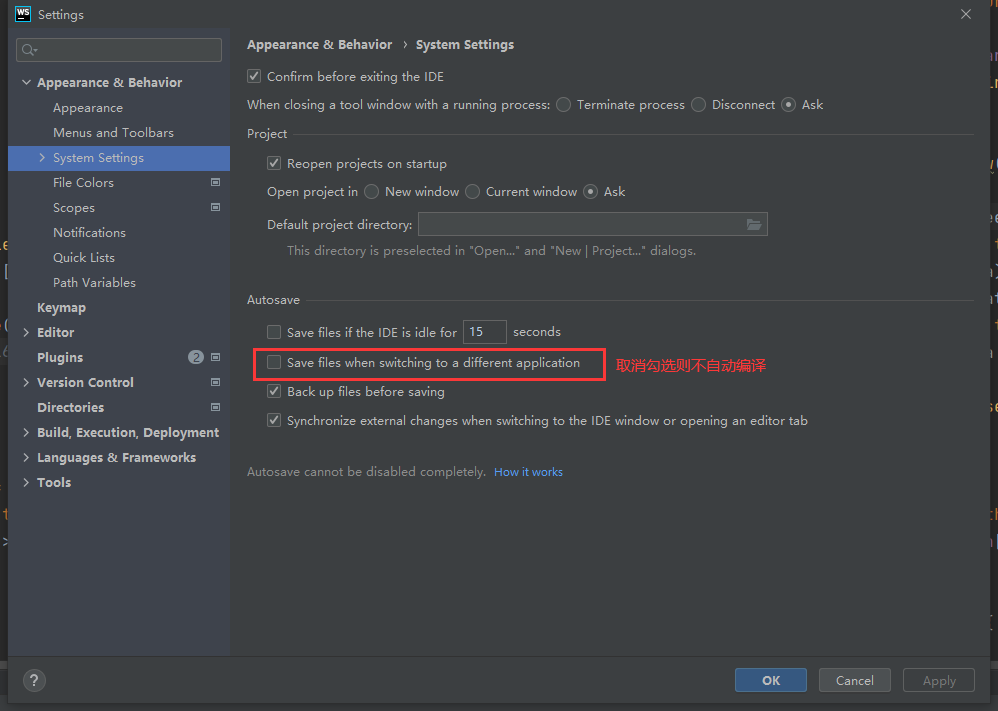
01.后台看我本地，而我又在改代码，实时保存编译的话，刷新页面了，他不方便看我本地；

02.项目太大，没改一处，自动保存频繁导致开发工具崩溃

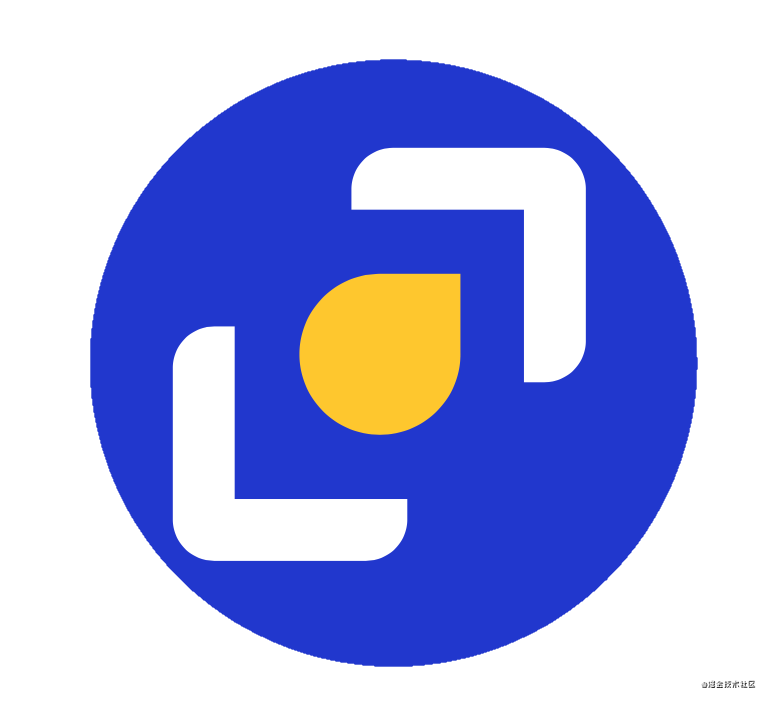
Ineffective mark-compacts near heap limit Allocation failed - JavaScript heap out of memory

打开setting设置，Crtl+Alt+S快捷键，勾选为自动保存

在红色方框内打勾，表示自动保存，取消则不自动保存。



# **都 2021 了你居然还在用 WebStorm ？**

[](https://juejin.cn/team/6948236603978416165/posts)

**[飞猪前端团队](https://juejin.cn/team/6948236603978416165/posts)**[IMG_257](https://juejin.cn/team/6948236603978416165/posts)

2021年05月19日 11:17 ·  阅读 29550

 关注



[飞猪前端团队](https://juejin.cn/user/3051900006845944" \t "https://juejin.cn/post/_blank)**[IMG_260](https://juejin.cn/user/3051900006845944" \t "https://juejin.cn/post/_blank)**公众号 @ fliggyfe

## 〇、起

都 2021 年了，你居然还在用 WebStorm？VSCode 不香吗？

****敲代码用啥只有一个条件就是你可以超级熟练使用那个工具，重点是可提高你独有的敲代码效率，有些像「君子性非异也，善假于物也」这句话表达的观点****。

那么到底 WebStorm 更香在哪儿，周末花了个下午怼篇文章来开一个 “历史倒车轮“，给大伙介绍一下「如何用上 WebStorm 最新正版？」「如何让 WebStorm 好用不卡？」，以及「WebStorm 香的地方在哪里？」，****以下文章将单词**WebStorm**换成**IntelliJ IDEA**也合适****。

## 一、如何用上 WebStorm 最新正版？



****很建议初步使用的时候「用上**[WebStorm 30天](https://link.juejin.cn/?target=https://www.jetbrains.com/zh-cn/webstorm/" \o "https://www.jetbrains.com/zh-cn/webstorm/" \t "https://juejin.cn/post/_blank)**内免费」的福利，等到觉得好用再去考虑申请正式版本，防止造成不必要的浪费！****

### 1. 自费去官方购买个人版

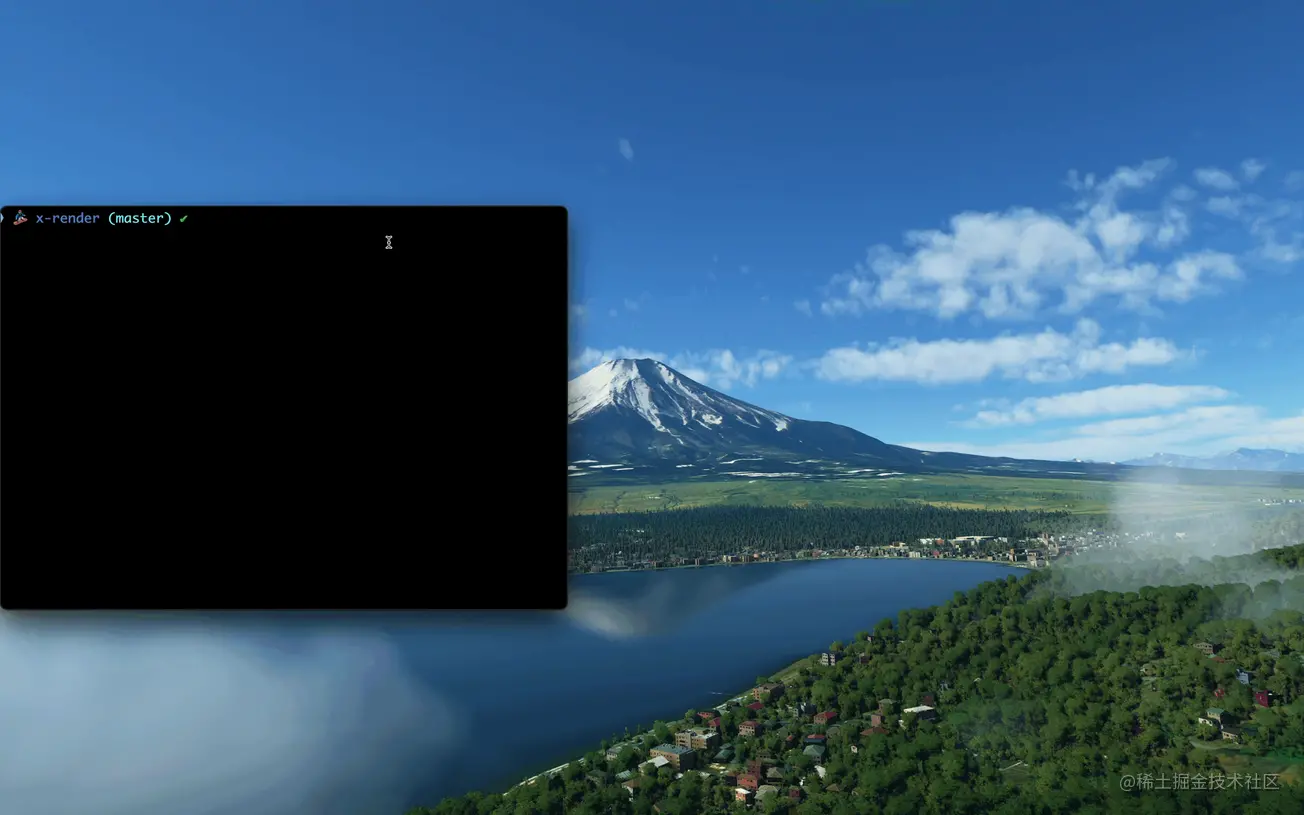
资金允许的同学可以直接上官网买[个人版本](https://link.juejin.cn/?target=https://www.jetbrains.com/zh-cn/store/%23personal" \o "https://www.jetbrains.com/zh-cn/store/#personal" \t "https://juejin.cn/post/_blank)，首年 59 美元，后面年续会便宜不少(59->47->35)，假如不更新的话还可一直用下去不续费。同时****对于个人版**[官方说](https://link.juejin.cn/?target=https://sales.jetbrains.com/hc/en-gb/articles/207241015-Can-I-use-my-personal-license-for-commercial-development-" \o "https://sales.jetbrains.com/hc/en-gb/articles/207241015-Can-I-use-my-personal-license-for-commercial-development-" \t "https://juejin.cn/post/_blank)**同样支持用于商业项目****，只需要保证一个人用即可。

### 2. 通过开源项目申请一年的使用

讲道理，JetBrains 公司真的很良心，提供了开源开发许可证福利，让大伙免费用他家软件进行开源贡献，一般项目维护在 3 个月以上大概率可通过。可从此入口 [免费许可证计划](https://link.juejin.cn/?target=https://www.jetbrains.com/zh-cn/community/opensource/%23support" \o "https://www.jetbrains.com/zh-cn/community/opensource/#support" \t "https://juejin.cn/post/_blank) 进入申请，也可参考 [这篇教程](https://link.juejin.cn/?target=https://zhuanlan.zhihu.com/p/87370573" \o "https://zhuanlan.zhihu.com/p/87370573" \t "https://juejin.cn/post/_blank)；除去开源申请外，还可试试[学术许可](https://link.juejin.cn/?target=https://www.jetbrains.com/zh-cn/community/dev-recognition/" \o "https://www.jetbrains.com/zh-cn/community/dev-recognition/" \t "https://juejin.cn/post/_blank)、[用户组](https://link.juejin.cn/?target=https://www.jetbrains.com/zh-cn/community/user-groups/" \o "https://www.jetbrains.com/zh-cn/community/user-groups/" \t "https://juejin.cn/post/_blank)、[开发者活动](https://link.juejin.cn/?target=https://www.jetbrains.com/zh-cn/community/events-partnership/%23developers" \o "https://www.jetbrains.com/zh-cn/community/events-partnership/#developers" \t "https://juejin.cn/post/_blank)、[开发者认可](https://link.juejin.cn/?target=https://www.jetbrains.com/zh-cn/community/dev-recognition/" \o "https://www.jetbrains.com/zh-cn/community/dev-recognition/" \t "https://juejin.cn/post/_blank)方式获取，****但都「不能用于商业项目」请斟酌使用****。

****综上，若资金允许就买个人正式版，有些紧就用开源项目的方案。**** 同时  来开发商业项目。

## 二、如何让 WebStorm 好用不卡？

[](https://link.juejin.cn/?target=https:/qpluspicture.oss-cn-beijing.aliyuncs.com/up/QYtuow.mp4)  
打开我 Mac 上最大的一个项目 [xRender](https://link.juejin.cn/?target=https://github.com/alibaba/x-render" \o "https://github.com/alibaba/x-render" \t "https://juejin.cn/post/_blank)，没有插电情况下，实际的启动效果 [录屏 mp4](https://link.juejin.cn/?target=https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/QYtuow.mp4" \o "https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/QYtuow.mp4" \t "https://juejin.cn/post/_blank)

### 1. 安装几个常用的插件

直接 command + ， 打开设置，找到 plugin，复制后在插件市场进行搜索安装：

1. ****中文语言包****：讲道理编程的同学应该默认全英文，但是安装后****你会一眼发现很多之前英语下“看不到“的东西****。想起之前自己强迫自己苹果全家桶都用英文，坚持了 1 个月就忍不下去了，虽然看得懂，但是需要脑袋 proxy 翻译的时间，回想起来何必呢？不过看英语如看中文无需反应的同学可忽视（等到哪一天功能熟练以后其实也可以关闭中文）。
2. ****Coderpillr Theme****：超级好看的一个主题，之前试过很多如 One Dark、Material Theme Ul 都不是很满意，其实最爱 Dracula Theme，但遗憾在这里不好看，刚好 Coderpillr 这个小众的很有 Dracula 风格，深得我心。
3. ****Key Promoter X****：快捷键爱好者的神器，会来智能教你记住快捷键。
4. ****.ignore****：自动提示你哪些文件可以被 ignore，省去新项目场景一个一个找的问题
5. ****WakaTime****：需要稍微配置一下，可以将你所有敲代码的地方装上这个插件，可以很访问的统计你敲了多长时间的代码，详细见[官网](https://link.juejin.cn/?target=https://wakatime.com/" \o "https://wakatime.com/" \t "https://juejin.cn/post/_blank)。

****对于插件，不建议安装太多，满足使用即可，太酷炫的反而影响性能，同时也建议禁用掉一批你用不上的****。

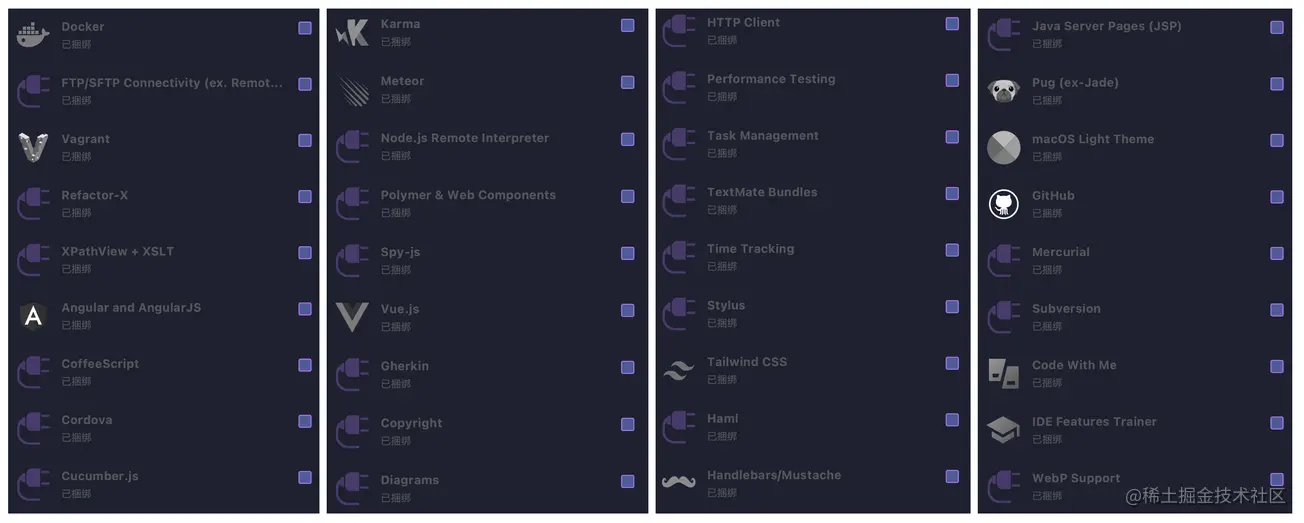
### 2.修改一下文字大小

主要有 3 个常用地方要修改下，可也按照个人喜好配置（[截图说明](https://link.juejin.cn/?target=https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/gatYqk.png" \o "https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/gatYqk.png" \t "https://juejin.cn/post/_blank)）：

* 一是外观字体，默认有些小了，建议 .ApplesystemUIFont 16；
* 二是编辑器字体，推荐 JetBrains Meno 20，行高 1.16，不要连体；
* 三是控制台字体，推荐 JetBrains Meno 18，行高 1.14，紧凑一些，不要连体；

### 3. 禁止掉不用的插件

继续在 plugin 中已安装 Tab 中，对你觉得你用不上的取消勾选禁止掉一批可能用不上的，减少使用内存，我禁止了很多，其中「Code With Me」不用也可以关闭，可以在「帮助->诊断工具->分析插件内存」来[看到](https://link.juejin.cn/?target=https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/qqeGgE.png" \o "https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/qqeGgE.png" \t "https://juejin.cn/post/_blank)更多可优化的空间，选择性参考如下：



### 4. 将默认内存设置大一些



1. 当前占用内存，可在底部栏看到，如果没有，可以右键底栏选择打开，也可以关闭一些不用的状态显示。
2. 在顶部菜单栏找到「帮助->更改内存设置」(2)，M1 很牛逼不动也行，其他的更新到 4096 即可(3)。

### 5. 记住和设置常用的快捷键

建议从这里开始，打开本地有 Git 发布记录的前端项目来测试，借此熟练一遍，建议在「设置->键盘映射」中，将「在 Touch Bar上 显示F1、F2等键」勾选上。由于正常软件为了最大化主编辑区，操作动作都在埋在菜单内部了，****导致不能所见所得从而想不到使用，记住快捷键可以让你没有想到情况下比普通用户用到多 50% 的潮流功能****。

****常用的一些好用快捷键，建议强迫多用最后到肌肉记住：****

**#***# 查找替换操作*

shift 按两次 // 随处搜索，搜索文件、功能、代码很方便

command + f // 当前页搜索

command + shift + f // 全局搜索字段

command + r // 替换当前文档

command + shift + r // 全局替换字段

**#***# 视图展示*

command + 1 // 显示隐藏左侧文件树

command + 0 // 显示当前待提交代码，用于提交前 diff

command + 9 // 显示当前历史所有的提交记录

command + 7 // 显示当前文件的结构，特别是看 Class 时候很方便

command + 上箭头 // 跳转到导航栏

**#***# 代码操作*

command + option + l // 格式化代码(常用)

shift + f6 // 使文件、标签、变量名重命名

f2, shift + f2 // 切换到上\下一个突出错误的位置

shift + 回车 // 无论在什么位置，自动跳到下一行

option + 回车 // 警告代码快速给出自动修正

command + 点击 // 跳到代码调用位置

command + delete // 删除当前行，command + x也可以

command + d // 复制新增一行一样的代码

command + w // 关闭当前文件选项卡

command + / // 注释行代码

command + b // 跳转到变量声明处

command + shift + c // 复制文件的路径

command + shift + [ ] // 选项卡快速切换，很有用

command + shift + u // 大小写自动转

command + shift + / // 注释块代码

command + shift + +/- // 展开/折叠 当前选中的代码块

command + shift + v // 从剪切板里选择黏贴

**#***# git 操作*

command + k // 填写代码提交记录

command + option + k // 提交代码

command + option + z // 撤销当前代码改动

command + d // 选中两个文件进行对比，充当文件夹 diff 工具复制代码

****找到「设置里的键盘映射」，进行操作文字「搜索」，设置几个有用的自定义快捷键，我习惯用左边竖直对齐的**control+shift**，也可以设置自己顺手的:****

* + 打开最近：设置成 control+shift+r 或者更顺手的 command+e，recent 可以将历史开发过的项目快速打开
  + 打开..：设置成 control+shift+o ，open 用于打开新项目，后面更多使用 command+e 快速启动
  + 关闭项目：设置成 control+shift+w ，用于关闭当前项目的 WebStrom 窗口
  + 终端：设置成 control+shift+i，iterm ****用于打开命令行窗口，这个命令用得很多****
  + 注解：设置成 control+shift+b，blame 可以显示当前文件代码每行的开发记录
  + 显示历史提交记录：设置成 control+shift+h，history 可以显示当前文件所有的提交改动记录
  + 拉代码：设置成 control+shift+p，pull 当前分支最新的的 git 代码
  + 与分支比较：设置成 control+shift+d，diff 可以将当前文件/文件夹（选择左侧的文件树）和历史分支对比，排查线上问题简直神器
  + 切换演示模式: 设置成command+option+control+v，view 特别合适在代码演示和分享给他人看的时候的场景
  + 切换免打扰模式: 设置成command+option+control+m，mute 特别合适想安静敲代码的时候

****刚开始不习惯使用快捷键的同学，可以在顶部菜单栏、在编辑器上、底部状态栏上、左侧导航文件树上多用用**右键**，可以发现很多新玩意，通过 Key Promoter X 来辅助记忆****

### 6. 可能用不上的高级设置

假如设置平时不卡，但是在 Webstorm 自带终端 npm i 以后，特别是依赖 node\_modules 文件索引超多变化导致无响应的问题（好电脑一般不会），可以试试我这边调试了好多次的高级设置。

在顶部菜单找到「帮助->编辑自定义属性」，打开 idea.properties 文件，编辑如下：

idea.cycle.buffer.size=4096

idea.max.intellisense.filesize=50复制代码

继续在「帮助->编辑自定义VM选项」打开 webstorm.vmoptions 文件，编辑如下：

-ea

-server

-Xms2048m

-Xmx8192m

-Xss16m

-XX:MaxMetaspaceSize=2G

-XX:MetaspaceSize=1G

-XX:ConcGCThreads=8

-XX:ParallelGCThreads=8

-XX:NewRatio=2

-XX:ReservedCodeCacheSize=240m

-XX:+AlwaysPreTouch

-XX:+UseG1GC

-XX:+DoEscapeAnalysis

-XX:+TieredCompilationUseG1GC

-XX:SoftRefLRUPolicyMSPerMB=50

-XX:+UnlockExperimentalVMOptions

-Dsun.io.useCanonPrefixCache=false

-Djava.net.preferIPv4Stack=true

-Dsun.io.useCanonCaches=false

-XX:LargePageSizeInBytes=256m

-XX:+UseCodeCacheFlushing

-XX:+DisableExplicitGC

-XX:+ExplicitGCInvokesConcurrent

-XX:+AggressiveOpts

-XX:+CMSClassUnloadingEnabled

-XX:CMSInitiatingOccupancyFraction=60

-XX:+CMSParallelRemarkEnabled

-XX:+UseAdaptiveGCBoundary

-XX:+UseSplitVerifier

-XX:CompileThreshold=10000

-XX:+OptimizeStringConcat

-XX:+UseStringCache

-XX:+UseFastAccessorMethods

-XX:+UnlockDiagnosticVMOptions

-XX:+HeapDumpOnOutOfMemoryError

-XX:-OmitStackTraceInFastThrow

-Djdk.attach.allowAttachSelf=true

-Dkotlinx.coroutines.debug=off

-Djdk.module.illegalAccess.silent=true

-XX:+UseCompressedOops

-Dfile.encoding=UTF-8

-XX:CICompilerCount=2

-Xverify:none复制代码

最后在顶部菜单栏找到「文件->清楚缓存..」，然后重启 WebStrom 后，再试试 npm i 安装巨多依赖应该也不会卡了。

当你上面都熟练了，很建议在右下角 [状态栏](https://link.juejin.cn/?target=https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/yXusL6.png" \o "https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/yXusL6.png" \t "https://juejin.cn/post/_blank) 中登录账号进行同步，防止设置丢失，等你用过一段时间后，你就会发现 WebStorm 慢慢的就香起来了。

## 三、WebStorm 香的地方在哪里？

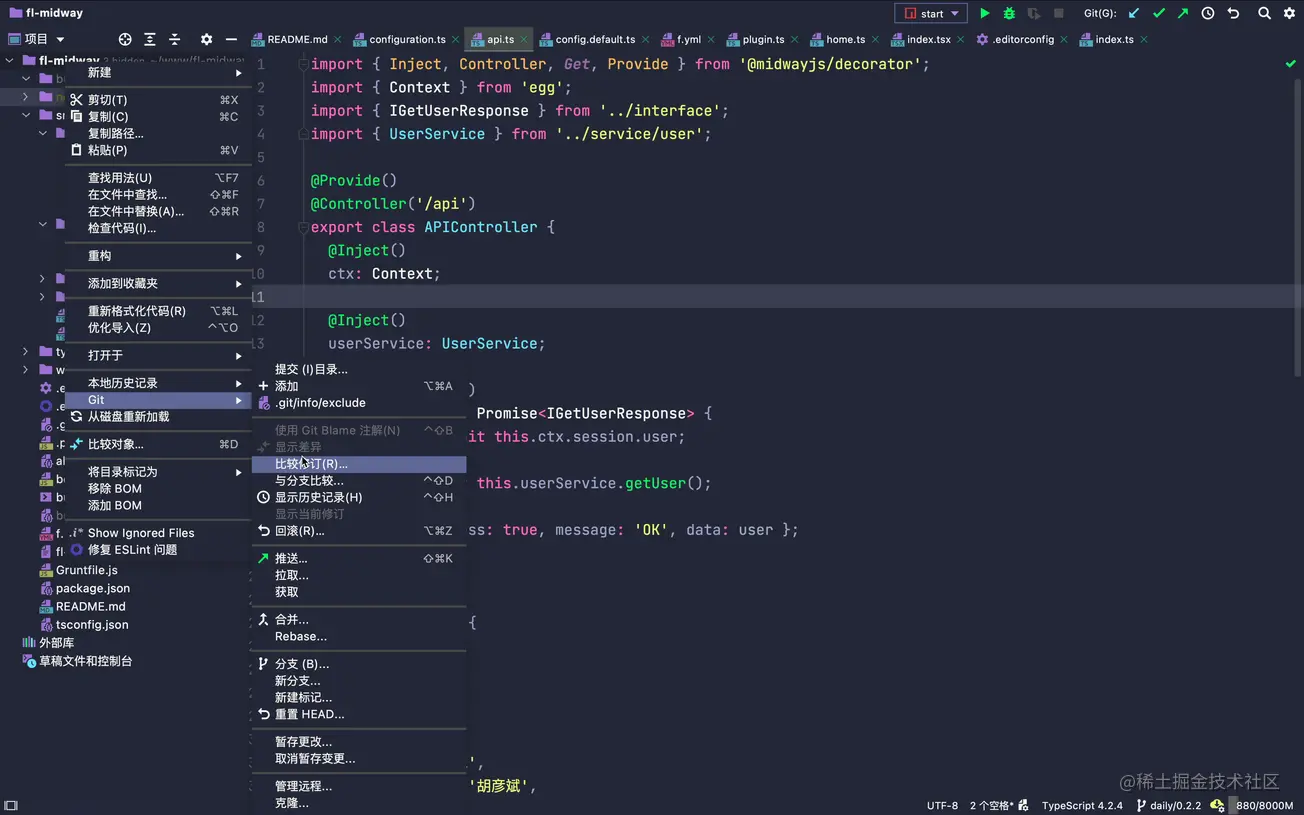
### 1. 世界上最好用 Git 管理工具

****看这一行代码是谁写的****：也叫 git blame，快捷键 control+shift+b，或者在代码行数数字式右键「使用 Git Blame 注解」，就可以出现了这样 [效果](https://link.juejin.cn/?target=https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/Ko1zcO.png" \o "https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/Ko1zcO.png" \t "https://juejin.cn/post/_blank)，同时可鼠标点击下对应的名称，会出现当时那次提交的 [详细记录](https://link.juejin.cn/?target=https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/60h8XE.png" \o "https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/60h8XE.png" \t "https://juejin.cn/post/_blank)。

****提交前 CR 代码改动****：很建议保持这个习惯，每次提交前都对比一下改动了下，快捷键为command + 0，这个 diff 对比超级清晰，同时很容易发现粗心的代码，这样的 [效果](https://link.juejin.cn/?target=https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/vgy10n.png" \o "https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/vgy10n.png" \t "https://juejin.cn/post/_blank)，确定无误后 command+k 填写提交记录，command+shift+k 提交代码，提交完成后 command+1 切回到代码开发。

****查看当前的所有提交记录****： 快捷键 control+shift+h 可以查看所有的历史提交记录，假如你只想看某个文件或者某个文件夹，选中，然后快捷键即可，或者右键找到 git 里面的「显示历史记录」。

****查看当前分支和主干的改动点****：这个是我觉得最好用的地方，也叫「****救命操作****」，比如说线上出现了问题，想对比上线版本和上一个版本的全部代码改动点就用得上他了，或者你需要合并 master 之前用于最后的 CR 也是很合适的，快捷键为 control+shift+d，或者在对应的总文件夹上面点击右键找到 git 里面的「与分支比较」,就可看到如下效果，同时可以按下 F7 来看下一个改动点。

  
有试过知名的 Tower、Sourcetree、Sublime Merge、Fork 等 Git 管理工具，发现真心不如它

### 2. 很安稳的重构&优化代码的神器

****文件、变量重命名会帮忙将所有的调用也修改掉****：快捷键为 shift+f6，也可以直接在文件或者变量上右键找到重构，改后所有地方都会帮你自动改动。

****文件修改位置后，也会自动修改调用处的引用****。

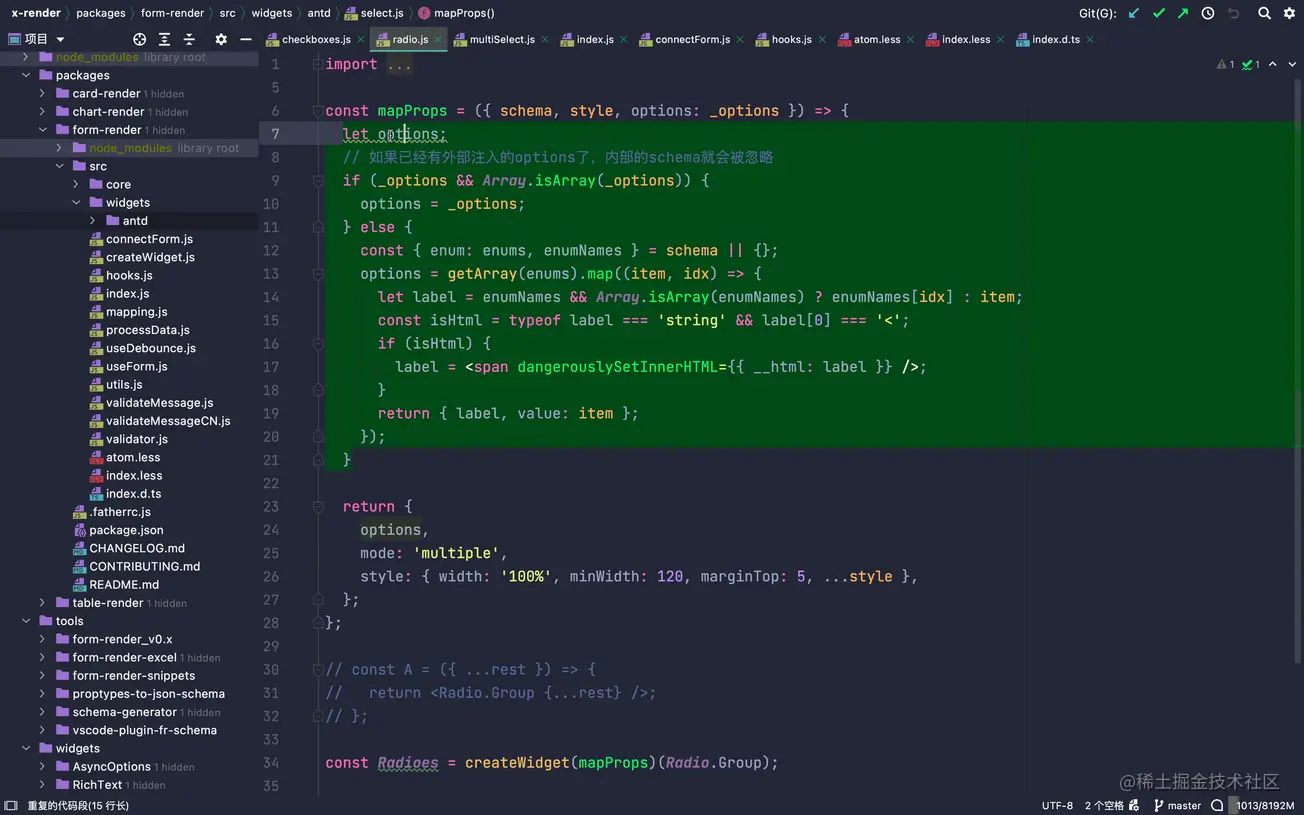
****文件删除前，会检测调用是否清理干掉****：可以在一个文件上 command+delete 试试，会提示你哪儿还在使用。

****拼写错误、语法错误、代码写得不优雅都会有提示****：可以 option+回车 来按照建议修复，有时候拼写错误可能是自有单词，可将此单词存到 WebStorm 字典里面。

****可以将某一块代码提炼成一个方法使用****：找到一块可以提炼成方法的代码，选中，然后 command+option+m 即可抽象方法并智能帮你方法命名。

****可替换成使用 Prettier 作为默认格式化****：这个功能可以更好的和其他同事 VSCode 开发场景中代码格式保持一致，可见 [设置演示](https://link.juejin.cn/?target=https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/06rBw9.png" \o "https://qpluspicture.oss-cn-beijing.aliyuncs.com/up/06rBw9.png" \t "https://juejin.cn/post/_blank)。

****告诉你哪些代码可以优化****：特别是重复代码的提示查找工具很是不错，此处偷偷 @NASA FR 有可优化代码。

1. 

### 3. 智能提示真的很强

****显示当前包信息****：可以将鼠标放到对应的引入包上面一会儿，会显示这个包的 README，省去去看如何使用：

1. 

****机器学习来提示代码****：写代码过程中，当前 ****WebStorm 2021 支持机器学习所有人的代码习惯来优化代码提示功能****，支持简洁的代码提示（VSCode 有时候会有很多用不上的提示），用了一段时间后你就慢慢发现了。

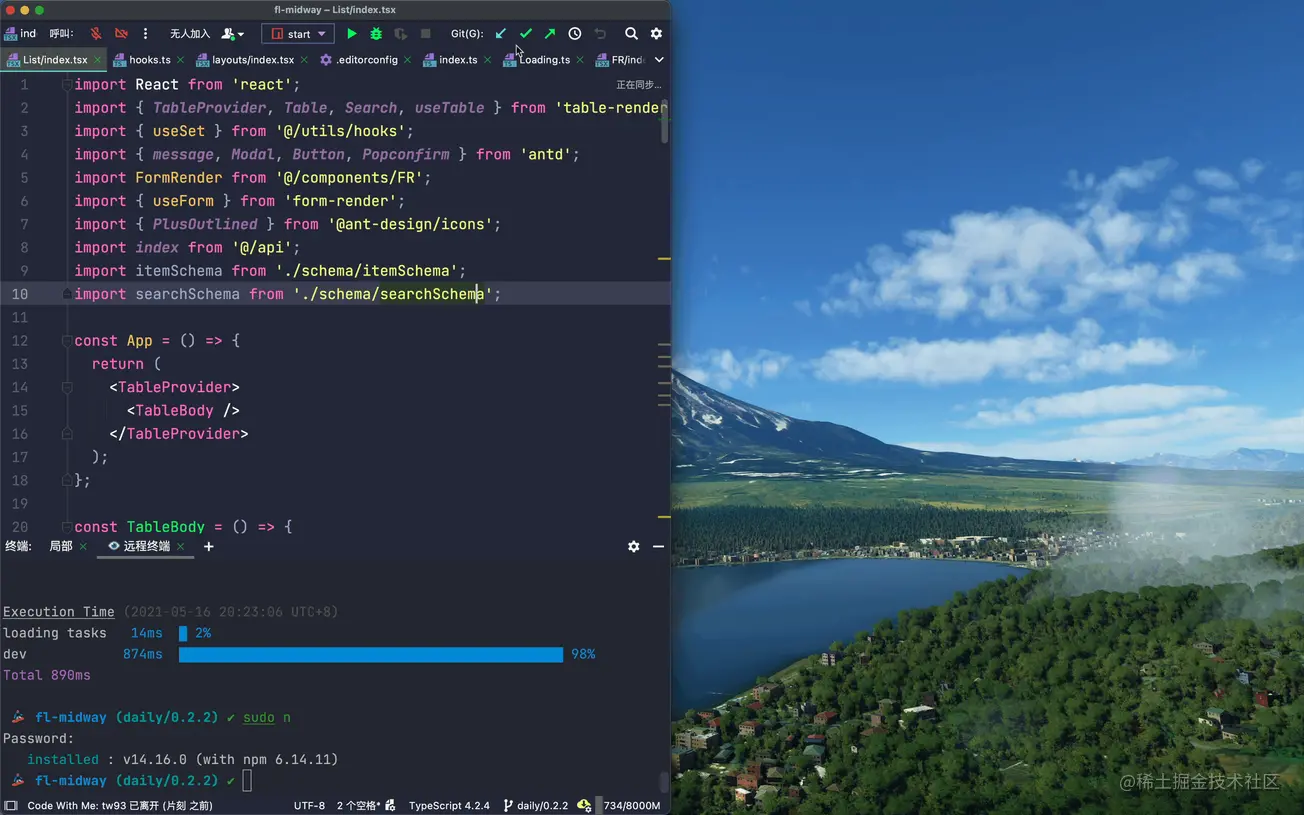
1. 

### 4. 潮流功能 Code With Me 实时调试他人的代码

回想起之前给同学排查代码问题，特别是不坐到一起的场景，你需要先让他加代码权限，然后 clone 代码，切分支，npm i，然后告诉你代码哪儿可能有问题，这里就浪费了很多时间了。

有时候可能会用 Mac 远程控制，但是有这里体验很不好，很多时候都是卡顿掉帧反应慢，寸步难行的感觉，更不要说去调试代码了。

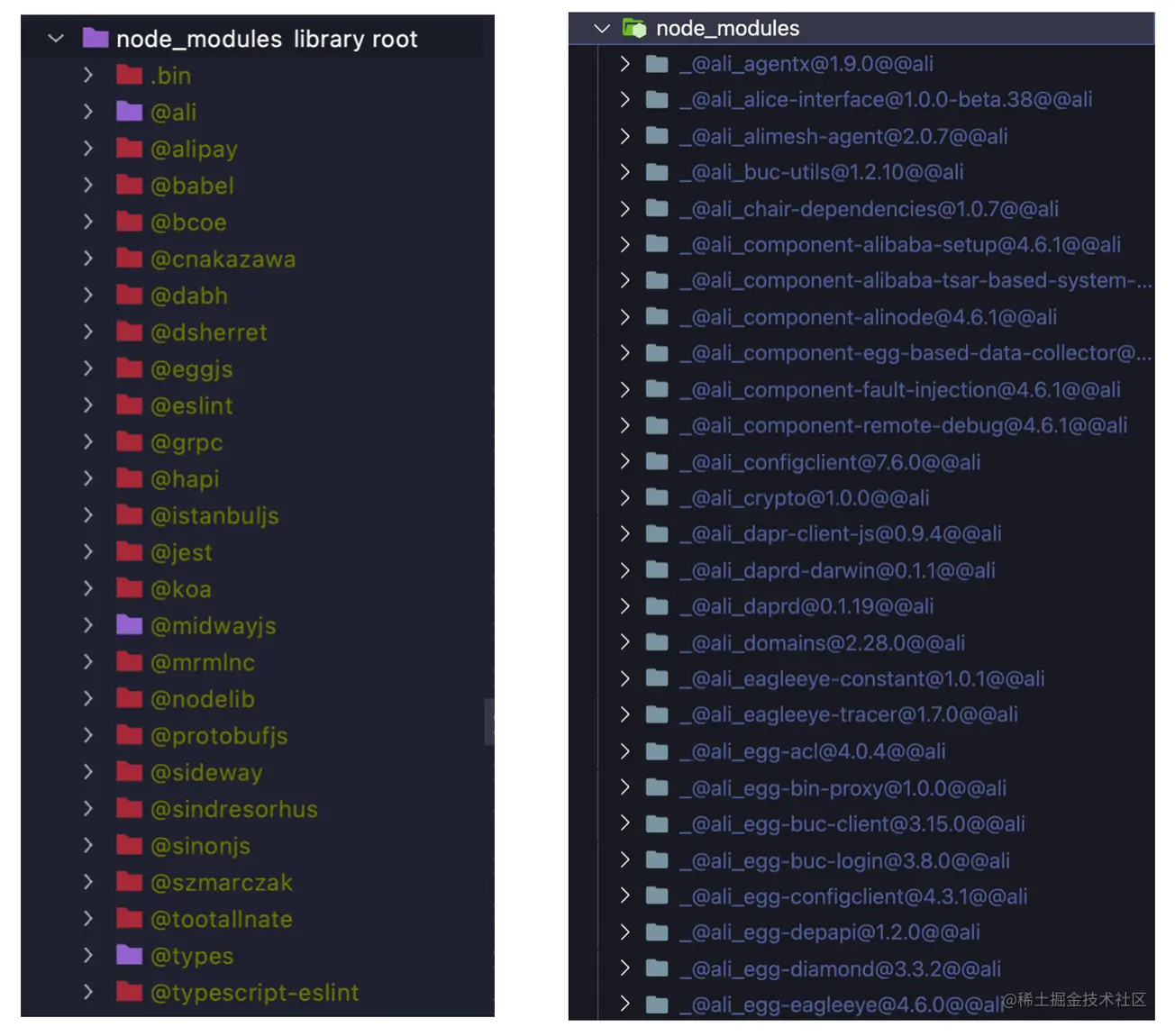
有了这个 Code With Me 后（没有开启需要去插件中去开启），直接对方分享一个 url 给你，会自动将对方代码包括安装好的依赖同步到你的 WebStorm，然后你在你本地修改时候，对方电脑也会修改，更潮流的你两还可以视频会议的方式来聊哪儿问题，使用几乎原生秒开的体验，可通过 [官方视频](https://link.juejin.cn/?target=https://www.youtube.com/watch?v=Lq0fCMCK-Yw" \o "https://www.youtube.com/watch?v=Lq0fCMCK-Yw" \t "https://juejin.cn/post/_blank) 了解更多。



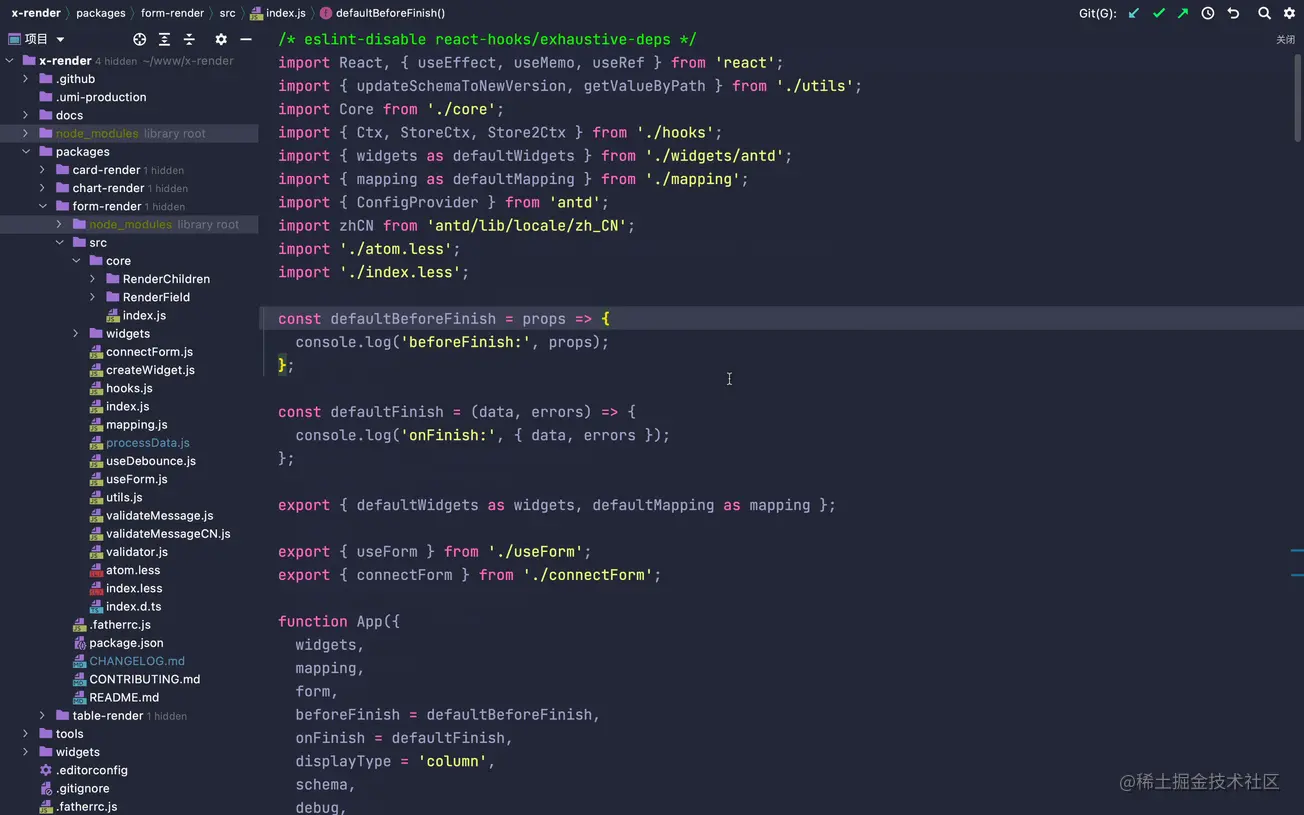
左边为本地 WebStorm，右侧为协助者的 Code With Me，类比两台电脑协同的效果

### 5. 还有不少小惊喜的功能

****给你一个清晰的 node\_modules****：可能很多用 VSCode 的同学都很困惑，我的 node\_modules 根本找不到东西，但是 WebStorm 就很清晰了：

  
左边为 WebStorm，右侧为 VSCode（不过可通过 Node Modules Context Menu 这个小众插件解决）

****独特的演示模式和专注模式****，可以在通过自定义快捷键 command+option+control+v 打开，或者在「视图->外观->进入演示模式」找到（可去设置->外观->最下面将演示模式字体设置到 28），省去演示代码时下面看不清的尴尬；使用自定义快捷键 command+option+control+v 进入免打扰模式，也可以「视图->外观->进入专注模式」认真敲代码，也是很清晰的。

  
第一次是演示模式，第二次是免打扰模式

****回到开始所说，提高操作效率不在乎你用哪一个工具，而在乎你熟练哪个工具，同时很建议大伙对于自己常用的软件，慢慢让自己习惯使用快捷键，并逐步脱离鼠标，会有有趣事情发生的****。

# **【缺缺缺人】杭州阿里飞猪招前端端端**

**[飞猪前端团队](https://juejin.cn/team/6948236603978416165/posts)**[IMG_257](https://juejin.cn/team/6948236603978416165/posts)

2019年04月09日 11:55 ·  阅读 896

 关注

[飞猪前端团队](https://juejin.cn/user/3051900006845944" \t "https://juejin.cn/post/_blank)**[IMG_259](https://juejin.cn/user/3051900006845944" \t "https://juejin.cn/post/_blank)**公众号 @ fliggyfe

### 团队介绍

我们是飞猪商旅前端团队，面向 2B 商务旅行业务，提供机票、火车票、酒店、租用车等相关预订、履约、逆向、营销服务能力承接，对接飞猪、淘宝、支付宝、钉钉等多客户端及 PC，提供混合式、高性能、富体验前端技术支持，同时围绕业务构建 Hybrid、离线化、Canvas/WebGL、小程序、Node 全栈等相关技术产品与平台，助力业务更好更快发展。

1）解决方案：H5 Hybrid 强混、Weex/Rax 跨端融合、面向多端的离线化体系、飞猪小程序、PWA、组件化、数据埋点/监控、Node Service/SSR、云构建、灰度/ABtest、Canvas/WebGL

2）技术产品：Clam 工程化套件（涵盖脚手架、开发/调试工具、发布管理等）、信鸽离线资源管理平台、页匠页面搭建平台、哈勃监控平台、Dragon Node 开发套件（涵盖Service、SSR、中后台等）、Janna 中后台开发套件、蜂鸟组件平台（包含 H5/Rax/iOS/Android/小程序 多端）、水母数据大盘（性能数据、监控数据、业务数据）、Weex-UI、互动平台（包含2D/3D 解决方案、微动效、素材库等）……

3）技术关键词：性能、稳定、沉淀、创新

### 职位要求

1. 本科或以上学历，3年以上前端开发经验
2. 精通各种前端基础技术，包括 HTML/CSS/JavaScript 等，熟悉函数式、性能优化、Hybrid 技术者优先
3. 熟悉至少一个主流框架（如 Vue/React/Angular 等），能够高效完成视觉稿拆解及编码实现
4. 对前端工程化与模块化开发有一定了解，并有一定实践经验（如 Gulp/Webpack/Rollup/CommonJS 等）
5. 对前端技术有持续热情，具备良好的团队协作精神，个性乐观开朗，善于与各种不同背景角色合作
6. 具备跨终端前端开发能力、具备客户端开发能力、有高质量开源项目、具备数据分析技能者优先

### 职位描述

1. 负责飞猪商旅相关业务的多端页面开发、上线、维护工作
2. 参与前端技术架构设计，承担核心模块的代码编写
3. 保障业务性能与稳定性，分析各类数据，识别并解决瓶颈问题
4. 在理解产品业务的基础上，打造提升产品体验或研发效能的技术工具、产品、平台
5. 关注并实践前端前沿技术，通过新技术反哺团队与业务

### 工作地点

杭州阿里

对阿里飞猪有兴趣的童鞋，可以把简历

发送至：**tw102972@alibaba-inc.com**

假如是后端 JAVA 同学看到此条消息，或者是你的小伙伴刚好在寻找机会，或者你的层级可以到P7，也可以将简历发送给 tw102972@alibaba-inc.com ，可以帮忙一起内推，谢谢大家！

# [Webstorm](https://blog.csdn.net/wb1046329430/article/details/116607344?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522165052398716780265434659%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=165052398716780265434659&biz_i)IntelliJ IDEA 之 Git 使用

