

## 15讲了解文件管理，什么是multi-rootworkspace



今天我们一起了解一下在 VS Code 中我们是如何管理文件和文件夹，以及如何在工作台中管理我们打开的编辑器。

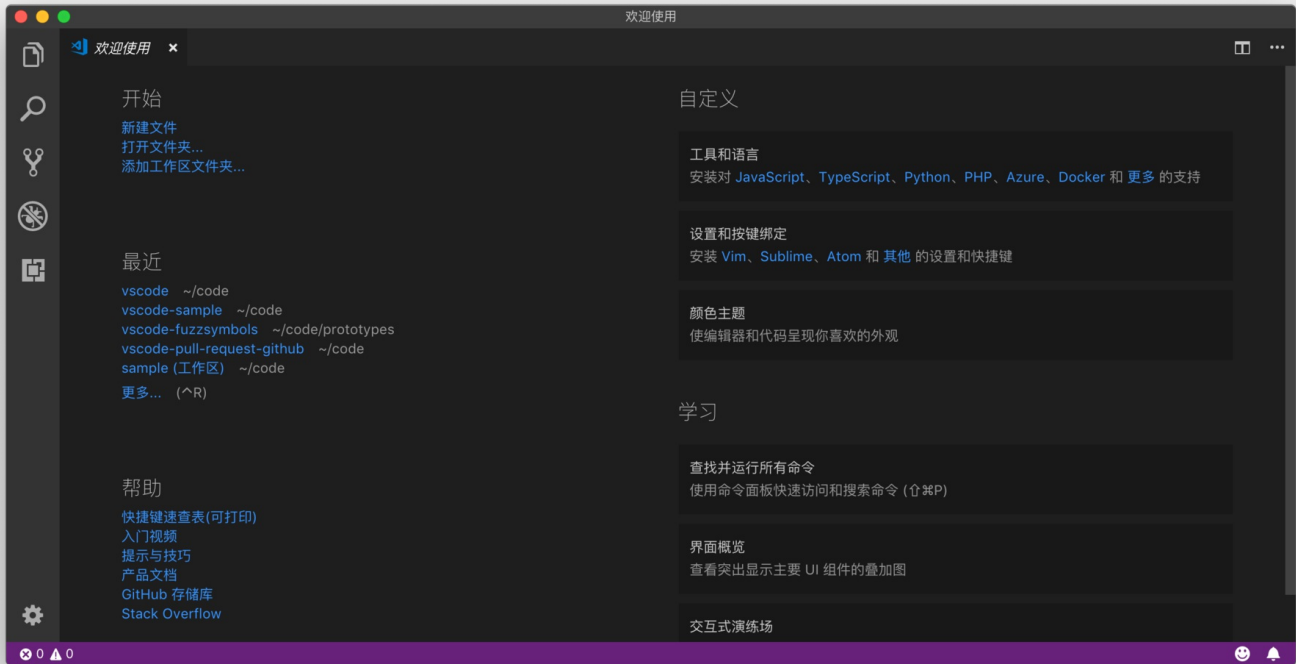
首先需要说明的是，VS Code 的各个功能，都是基于当前打开的文件或者文件夹的。

该怎么去理解这个概念呢？如果你使用过 IDE 的话，你应该记得在第一次打开 IDE 的时候，它们往往需要你创建一个工程，这个工程会生成一个特殊的工程文件。这个工程文件记载了这个项目有哪些相关的文件、项目的配置、构建脚本等等。这个文件记录着 IDE 管理工程的元信息，开发团队也能够通过共享这个工程文件保证成员工作环境的一致性。但是工程文件对用户体验就不太友好了，比如说项目文件可能对 IDE 的版本有所要求，项目文件损坏了 IDE 读取不了但是我们也不知道如何修复，等等。

VS Code 则选择了一种相对轻量，而且大家都易于理解的方式，那就是**所有的操作都基于文件和文件夹**。当你打开一个文件夹，VS Code 的核心功能就会对这个文件夹进行分析，并提供对应的功能。比如，在打开的文件夹下检测到有 .git 文件，就加载 Git 插件来提供版本管理的功能；或者发现文件夹下有 tsconfig.json，就会激活 TypeScript 插件提供语言服务。

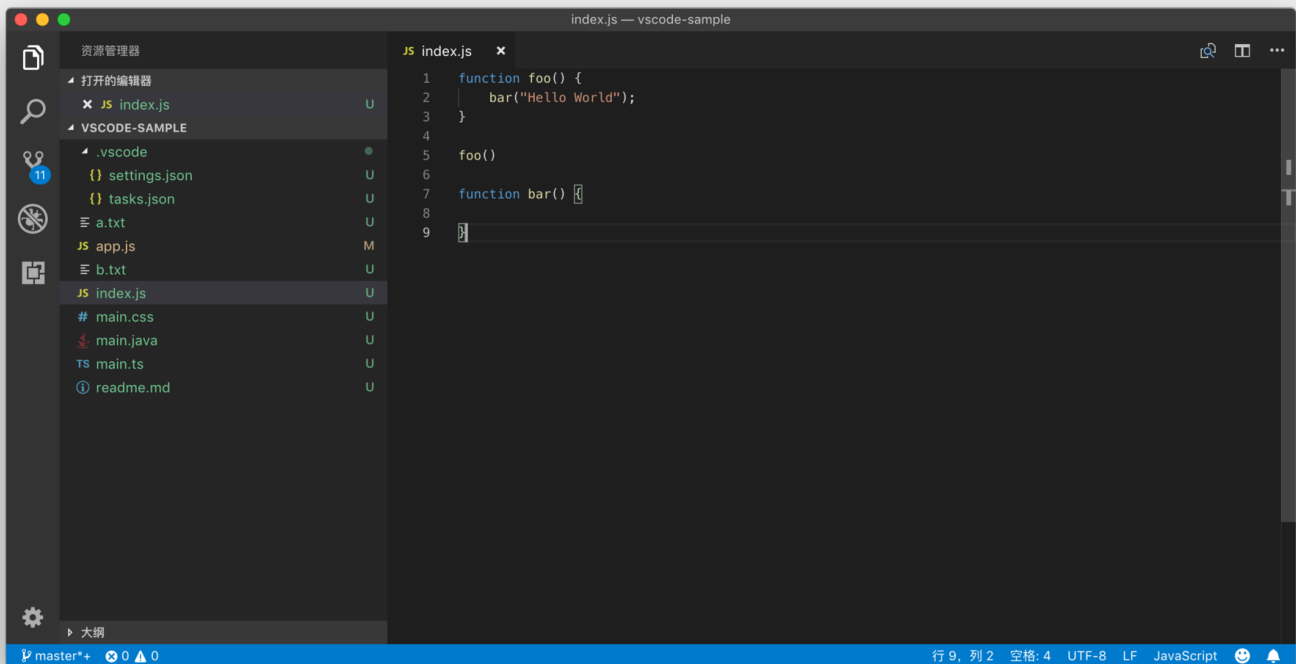
当你第一次打开 VS Code 的时候，工作台还没有打开任何文件夹。这时候在欢迎界面的左上方，你能够看到：“新建文件”和“打开文件夹”等这样的快捷键。

QQ: 1046877154, 微信: 110获取



未打开文件夹，状态栏为紫色

这时候请注意工作台最下方的状态栏，当 VS Code 没有打开任何文件夹的时候，它的颜色是紫色的。而如果在工作台中打开了某个文件夹，状态栏的颜色就会变成蓝色。



打开文件夹后，状态栏变成蓝色

## 资源管理器

当你打开一个文件夹之后，你就能在工作台的最左侧看到资源管理器。资源管理器将当前文件夹下的文件和子文件夹，以树形结构的形式呈现出来。每一个文件的前面还会有一个小图标，用于反映文件的类型。

在这个列表里，你既可以使用鼠标来打开文件，展开文件夹，也可以通过鼠标右键来唤出上下文菜单，对文件或者文件夹进行重命名、删除等操作。另外，资源管理器还支持鼠标进行多文件选择和拖转（drag & drop）。相信这已经是大家都非常熟悉的一种设计语言，这里我就不多加赘述。

在资源管理器的最上方，你可以看到一个列表，叫做“打开的编辑器”。这个很好理解，就是指这里面列出的都是当前已经被打开的文件。

文件列表的顶部，有一个标题栏，上面显示着当前打开的文件夹的名字，在我们文中的图片里，你可以看到这个文件夹的名字是“VSCODE-SAMPLE”。如果你把鼠标移动到标题栏上时，就能够在文件夹名的右侧看到几个按钮。



资源管理器文件列表标题栏的按钮

然后，你只需把鼠标移动到这些按钮上，就能看到它们的名字，分别是新建文件、新建文件夹、刷新和全部折叠，具体功能想必你一看即可知之。同样的，这些操作也可以通过右键上下文菜单来唤出。

## 设置

资源管理器你应该比较容易熟悉，也许不用介绍你也可以很快就能琢磨出来，不过这里还有几个设置需要我分享给你，以后在使用编辑器的过程中可能会用得着。

第一个是“explorer.autoReveal”，默认是打开的。当你在编辑器里打开一个文件时，资源管理器里的列表会自动将选中这个文件，并且滚动列表，将这个文件项滚动到屏幕的中间，保证你能够看到。如果你不喜欢这个功能，可以将其关闭。

第二个是“explorer.confirmDelete”，默认也是打开的。当你在一个文件项上右键选择删除时，VS Code 会跳出一个确认窗口，询问你是否确实要删除此文件。这个询问的目的是防止你误操作，想必这个功能你已经很熟悉了，当然，如果你觉得它多此一举的话，也可以将其关闭。

## .vscode

看到这里你可能会问，VS Code 是基于文件夹来进行管理的，各个插件也会分析当前打开的文件夹，但这是不是也意味着 VS Code 其实对项目没有任何管理和配置的能力呢？

其实不然，VS Code 是允许你创建几个跟当前文件夹或者跟项目有关的配置保存在这个文件夹中。这跟 IDE 的工程文件的思

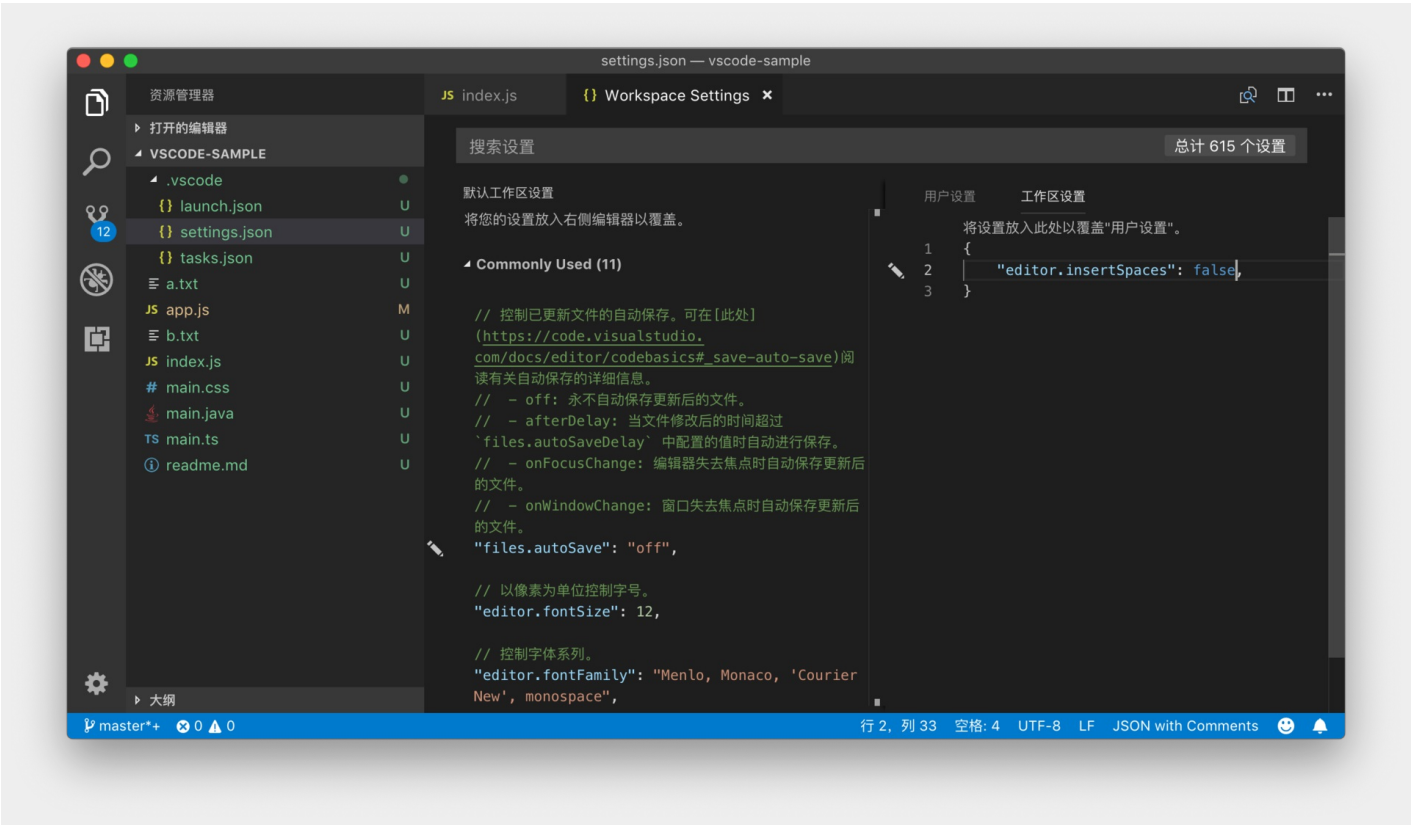
路是一致的，把设置放在这个文件夹中，便于团队内部进行共享。但二者区别在于，VS Code 的绝大部分功能，不需要这些配置文件也能工作，同时，这些文件都尽可能保证易于阅读和修改。

为了和其他工具添加的配置文件区分开，VS Code 的配置文件都会放在一个子文件夹 .vscode 中，下面我们一起来看这个特殊的文件夹。

这个文件夹中可以包含以下几种文件。

第一个是配置文件（settings.json）。这个文件的作用和格式，跟我们之前修改的用户设置几乎是完全一样的。唯一的区别就在于这个文件的设置，只有当前这个文件夹在VS Code 中被打开时才会生效。

这个文件的存在，很好地解决了个人喜好和项目规范之间的冲突。比如说我写代码的时候喜欢用空格，而不是制表符，这样的话，在我的个人设置里面就可以设置为永远使用空格。但是在VS Code的这个项目中，团队要求一定要都使用制表符，那我们就可以在 settings.json 中将这个配置写进去，而不是粗暴地要求每一位成员都去修改他们各自的配置文件。



另外常用的两个文件：一个是调试设置（launch.json），它是用于说明如何调试当前文件夹下的代码；一个则是任务设置（tasks.json），它是关于 VS Code 任务系统的配置文件。把它们放在文件夹中甚至一并放入到代码仓库中，这样任何使用这个项目的工程师，都不用学习如何配置调试和配置任务系统就能运行代码了。

当然肯定也会有人不喜欢在代码仓库中出现编辑器相关的各种配置文件，不过也不用担心，你可以使用版本管理的忽略配置，将 .vscode 这个文件夹剔除在版本管理之外。那么，你就可以自己去控制跟这个项目相关的配置文件了。

### 多文件夹工作区（multi-root workspace）

今天我要介绍的第二个主要知识点叫做多文件夹工作区（multi-root workspace）。老实说呢，这个概念是有一定的理解难度的。

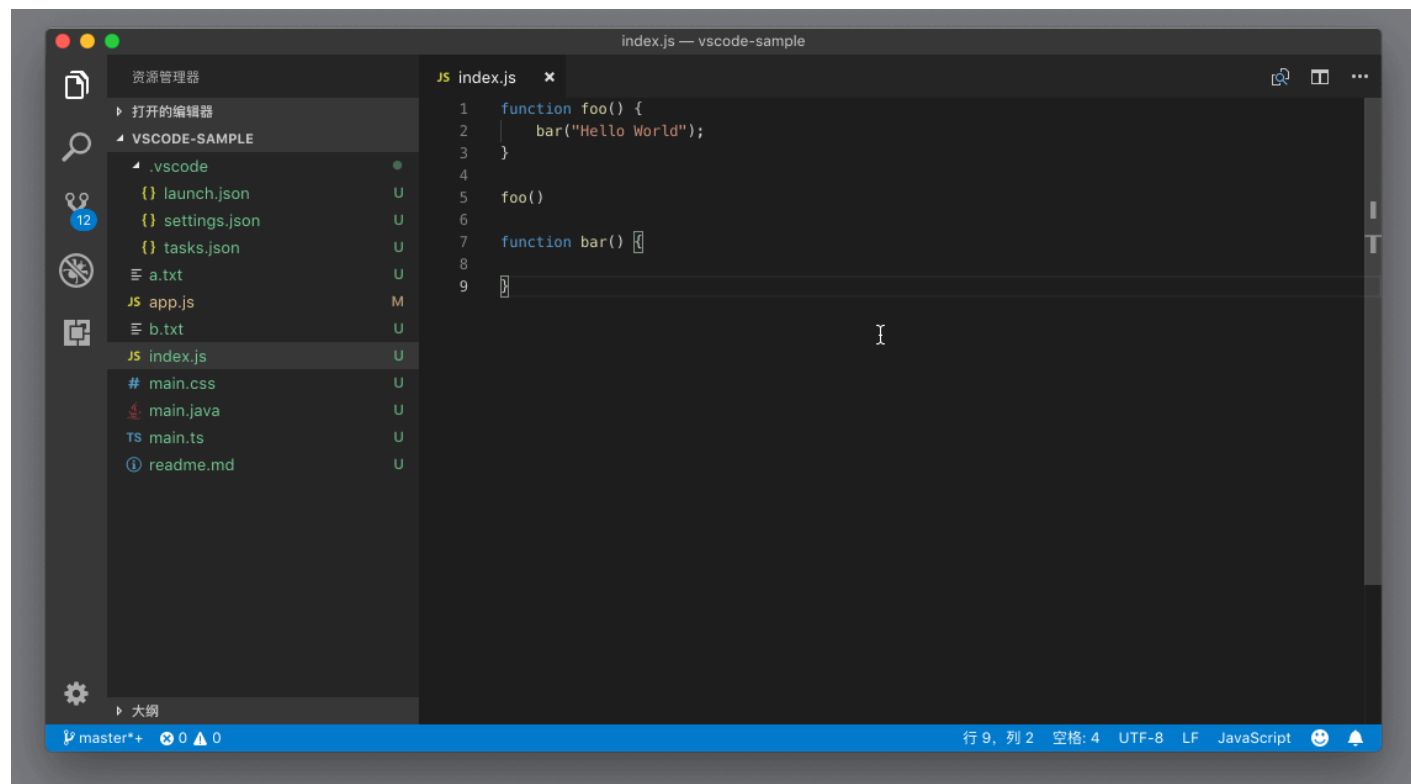
上面我们提到的基于文件夹的这种项目管理方式，从 VS Code 第一天开始就存在了。也几乎从第一天开始，我们就收到了用户对于这一个设计不满的反馈。对于这些不满的用户而言，他们的痛点在于他们经常需要同时对多个文件夹下的代码进行操作



作。但是 VS Code 关于单个文件夹的这种操作模式，要求了他们必须同时打开多个窗口，并不停地在它们之间切换。

多文件夹工作区就是为了针对这个问题而实现的解决方案。那下面我们就一起来看一看怎样去创建一个多文件夹工作区。

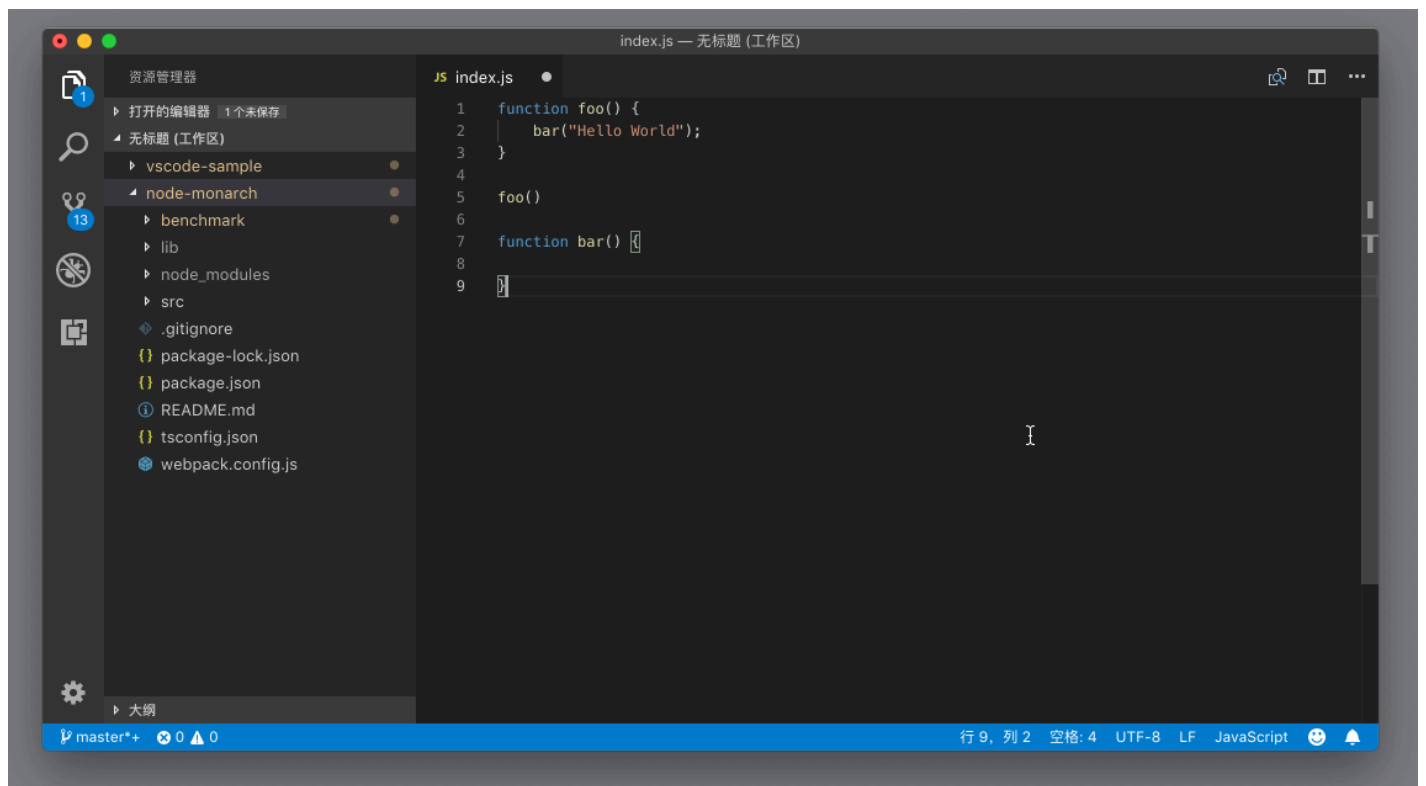
首先，在 VS Code 中打开一个文件夹，此时 VS Code 处于一个单文件夹的状态。然后你可以调出命令面板，搜索“将文件夹添加到工作区”（add folder to workspace）并执行，或者使用菜单，“文件 -> 将文件夹添加到工作区”，这之后，选择你当前窗口打开的文件夹。



### 添加新的文件夹到工作区内

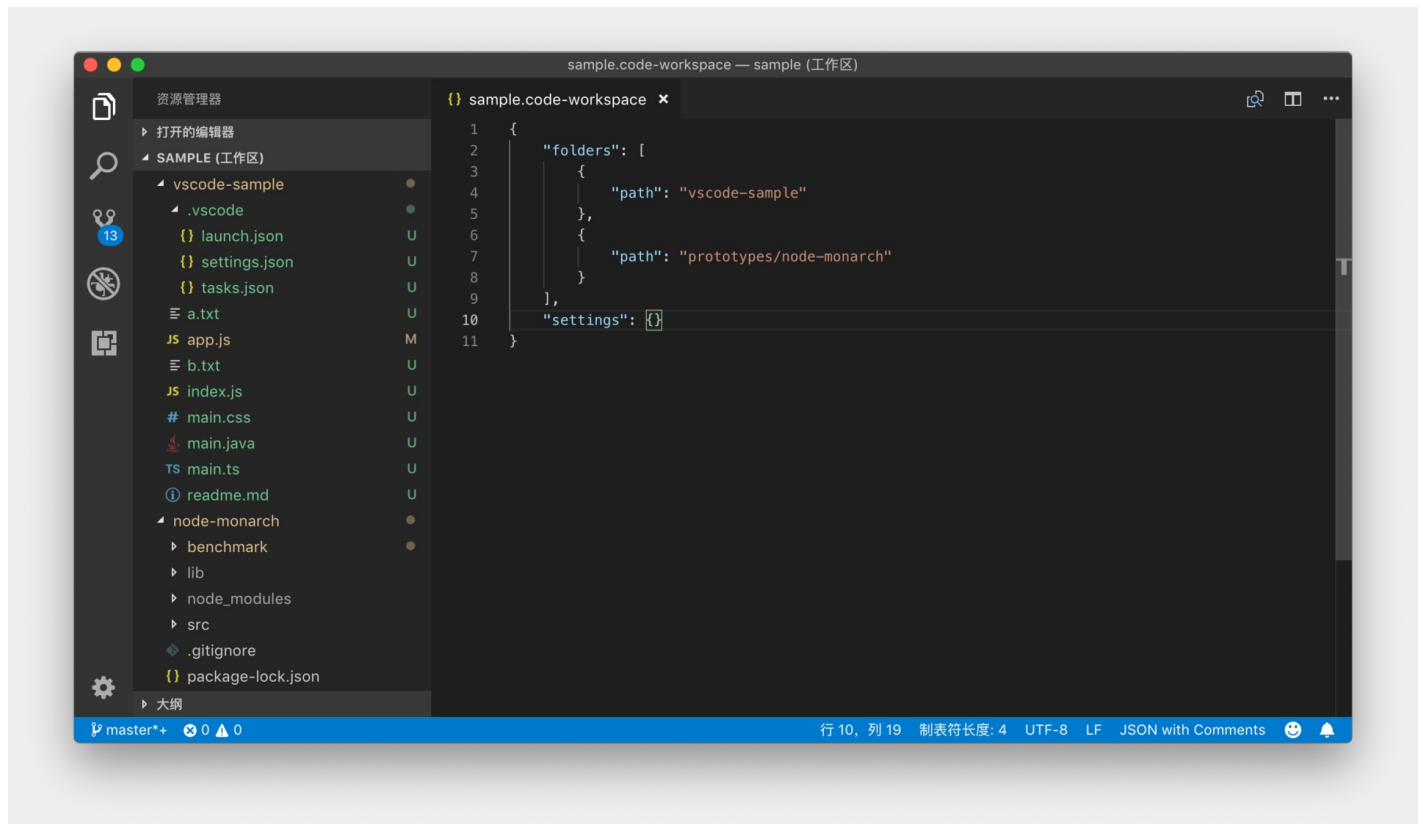
此时在资源管理器里的标题栏里，你能看到“无标题（工作区）”这样的文字，这说明当前的工作区已经有多个文件夹了，只是现在你还没有保存这个多文件工作区，也没有给它指定一个名字。

要保存这个工作区，接下来你可以调出命令面板，搜索“将工作区另存为”（save workspace as），VS Code 就会为这个工作区创建一个文件，这个文件的后缀名是“code-workspace”。比如，在下面的动图中，我给这个工作区取名为 sample，然后指定在 Code 中这个文件夹下保存。这样操作后，VS Code 就会在 Code 文件夹下创建一个 sample.code-workspace 文件。



### 保存多文件夹工作区

你可以看到，操作完之后资源管理器的标题栏已经相应地改变了。另外，sample.code-workspace 虽然有个特殊的后缀，但这个文件的格式其实也是 JSON，你可以自行打开这个文件查看一下。



打开sample.code-workspace这个文件

```
{
  "folders": [
    {
      "path": "vscode-sample",
    },
    {
      "path": "vscode"
    }
  ],
  "settings": {}
}
```

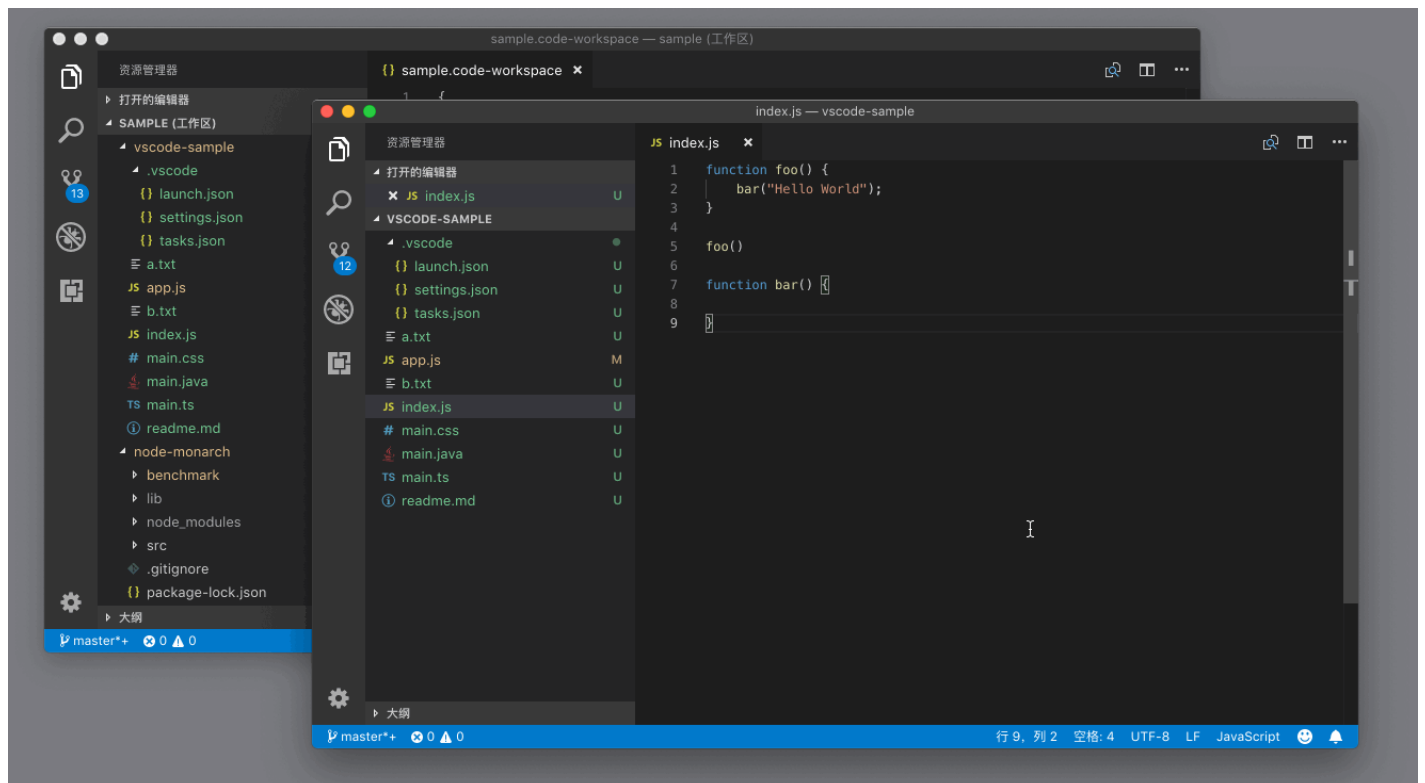
这个 JSON 文件，默认有两个键（key）。第一个是 folders 文件夹，它里面罗列的是这个多文件工作区里有哪些文件夹。可以看出，这些文件夹的地址，都是这个 sample.code-workspace 文件的相对路径。第二个则是 settings 设置，你可以在这个值里面添加专属于这个多文件夹工作区的设置。它的作用，跟上面我们介绍的 .vscode 文件夹下的 settings.json 文件是类似的。

此外，这个文件还有两个可选的键，它们分别是 extensions 和 launch，相信看到名字，你已经猜出来它们大概是跟什么相关的了。这部分我会在后面的章节中详细介绍的。

## 工作区切换

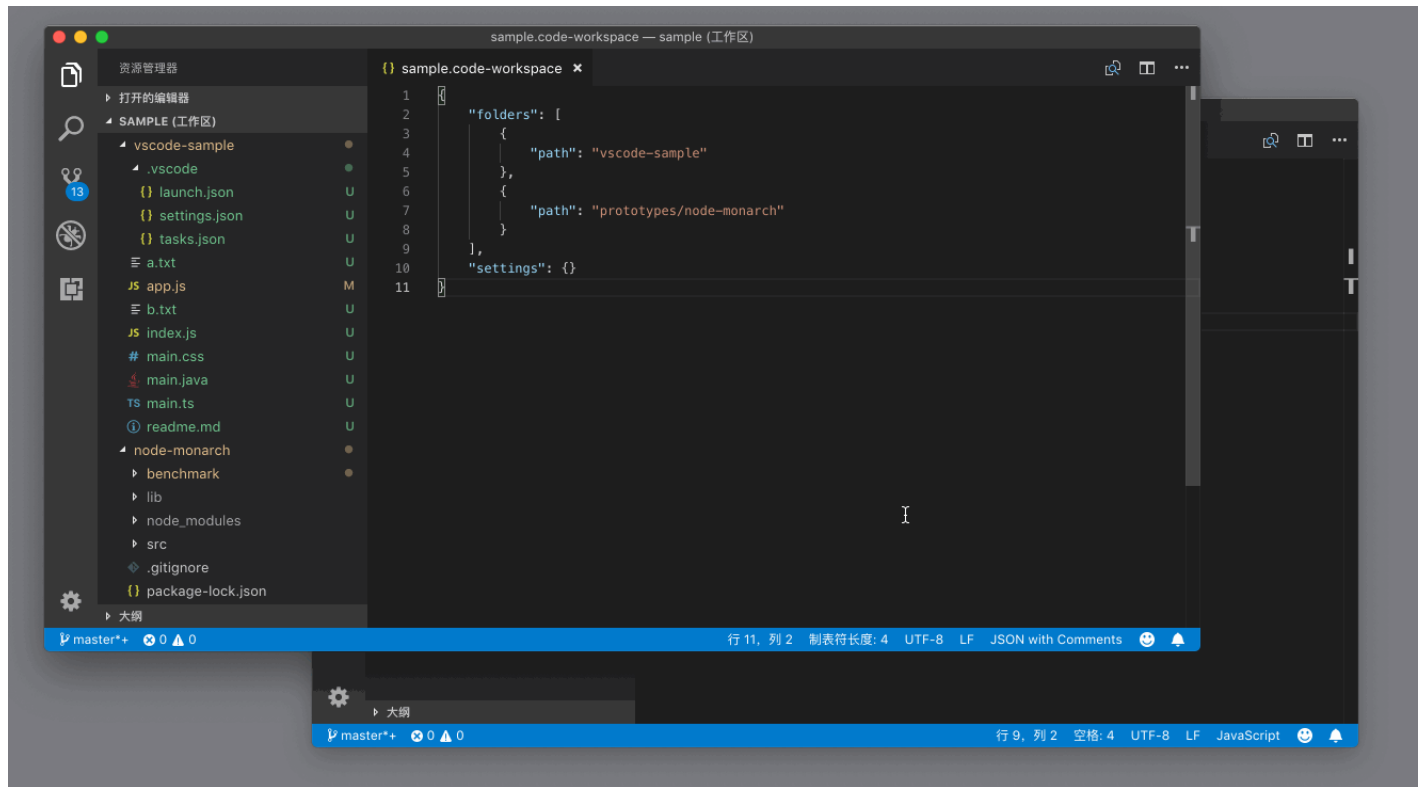
上面就是 VS Code 关于文件夹管理的主要知识了。刚开始我说多文件夹工作区有点复杂，你可能并不这么觉得。我有这种感觉主要还是跟我的个人使用习惯有关，我喜欢为每个文件夹创建一个新的窗口，在这些窗口快速切换本身就已经很方便了。

如果你同时打开了多个窗口，可以按下 Ctrl + W，或者调出命令面板，搜索“切换窗口（Switch Window）”，然后选择你要跳转的那个文件夹中去。



## 切换窗口

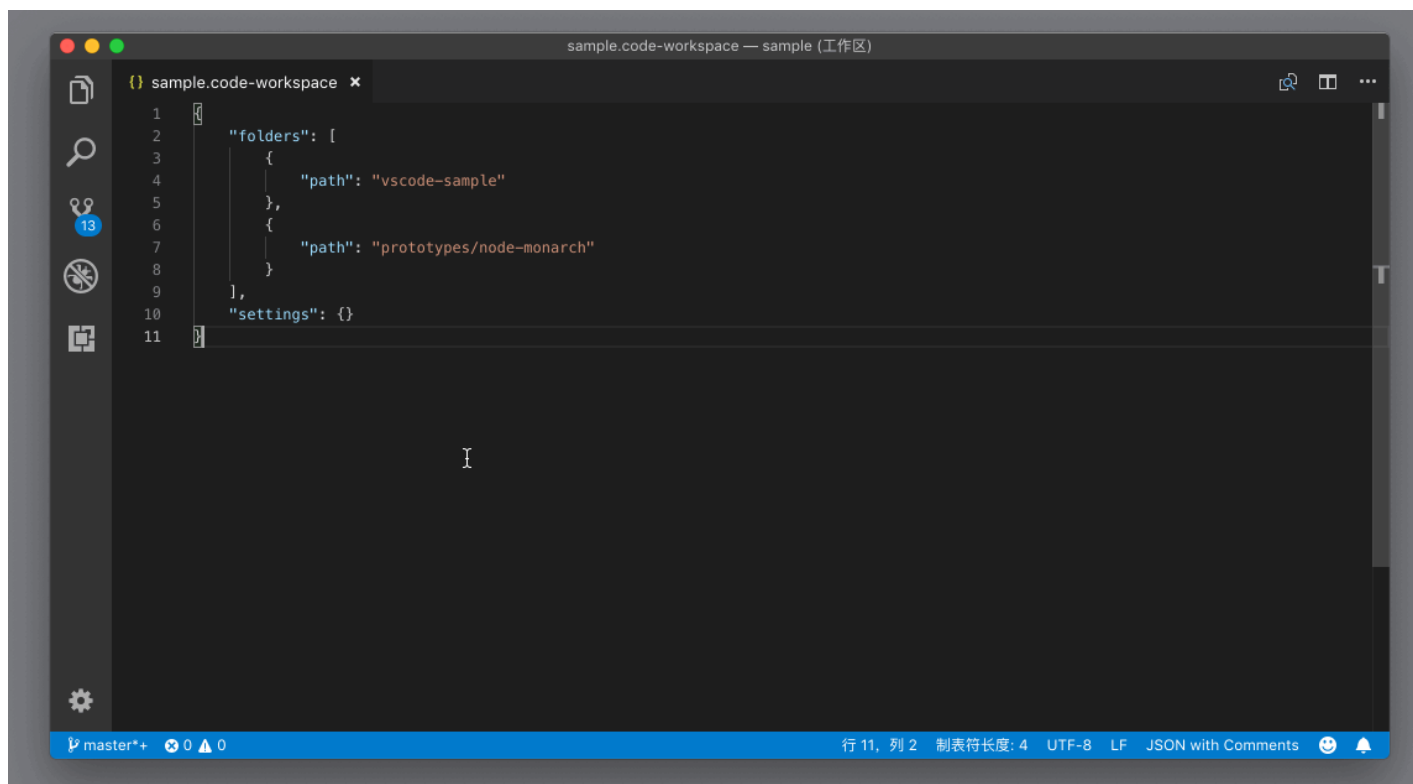
如果你只是要跳转到上一个打开的窗口，那就更方便了。打开命令面板，搜索“快速切换窗口（Quick Switch Window）”并执行，就能够直接跳转到之前的窗口了，而无需再做选择。这里我倒是非常建议你给这个命令指定一个快捷键，这样你就能在窗口之间一键切换了。



## 快速切换窗口

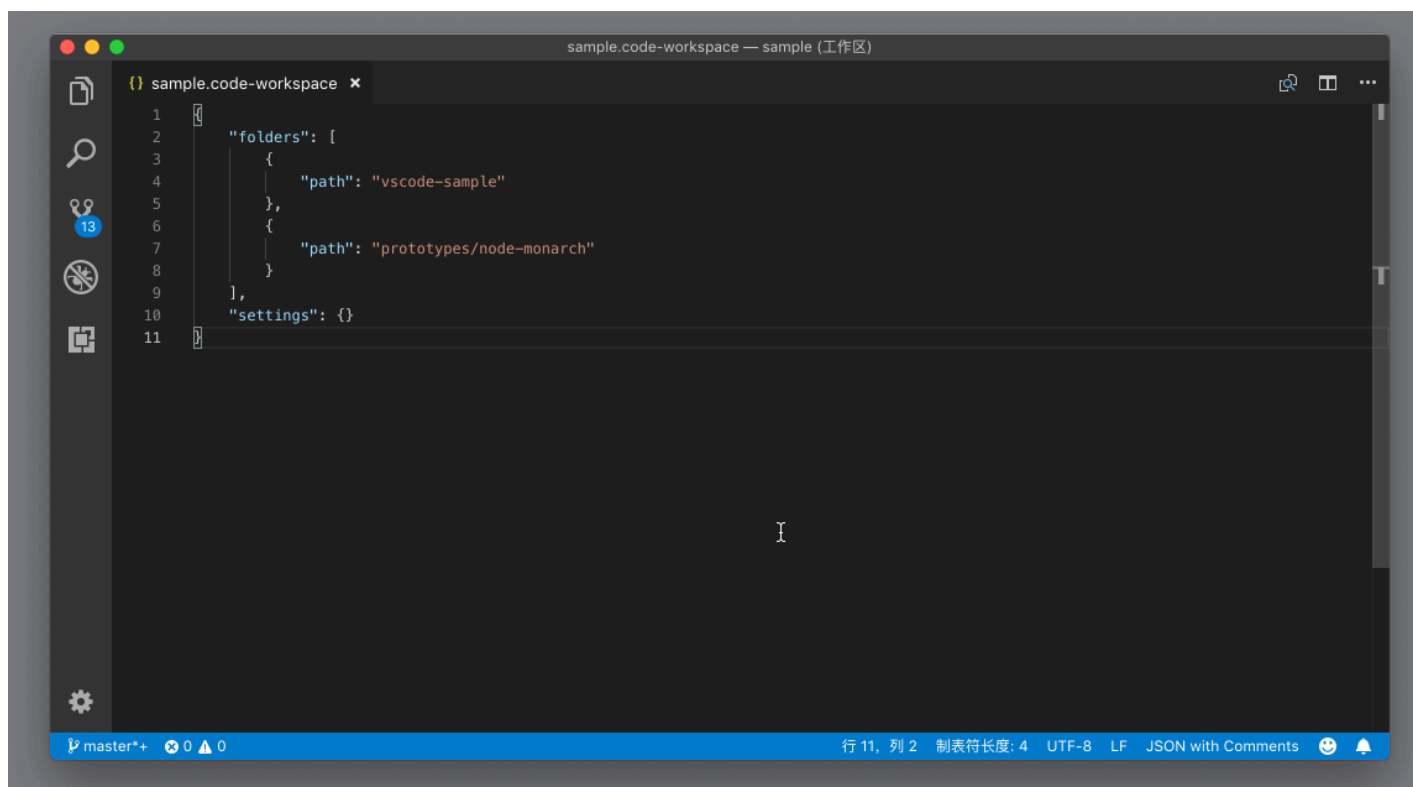
如果你同一时间只会关注一个项目，那你也大可不必使用多个窗口。我就经常只用一个显示器和一个窗口，然后当我想在另外一个项目上工作时，我就会按下 Ctrl + R（或者使用命令面板，搜索“打开最近的文件”），此时我就能够看到最近操作过的文件夹并按下回车键进行切换了。





### 切换文件夹

当你按下 `Ctrl + R` 调出最近打开的文件夹的列表后，也能够按下 `Cmd + 回车键`，将它在一个新的窗口中打开。



### 将文件夹打开在新的窗口中

正是因为有上面这几个命令的存在，让我觉得没有多文件夹工作区也是可以的。当然，多文件夹工作区在某些方面的优势是不可比拟的，比如说跨文件夹的代码调试，这个我们后面也会介绍。

## 小结

这就是我们今天的全部内容了。你可以试着重新审视一下，你是如何管理你的项目、文件夹的？能不能通过 VS Code 的多文件夹系统和快捷命令们对它们进一步优化呢？或者你有什么好的经验或技巧，也欢迎你留言和分享。



### 精选留言



Trust me 🐼  
mac cmd+` 可以切换窗口  
2018-10-17 00:37



啾啾啾朱  
大神，关于个人setting.json的设置能不能详细的讲下呢？没看明白  
2018-11-13 10:14



冰山北极熊  
老师，个人设置是在哪里配置？不打开文件夹进行配置就是个人配置吗？  
2018-10-17 12:02



冰山北极熊  
老师，配置文件怎么区分是个人配置和项目配置呢？是需要每个项目都要进行配置吗？  
2018-10-17 11:47

作者回复

在文件夹下的是属于项目的配置，如果你希望每个项目都总一样的配置，那么放到个人设置里即可  
2018-10-17 12:00



一步  
将多个项目 添加到同一个工作区，保存为 code-workspace 文件后，怎么把这个code-workspace文件也添加到工作区呢？，这样可以可以方便查看设置,麻烦老师解答一下呢  
2018-10-16 10:27



谢mingmin  
问题是,搜索的时候，也是搜索整个工作区。如果可以设置只搜索某个项目文件夹里面的东西会比较好一点。至少速度快些  
2018-10-16 10:08