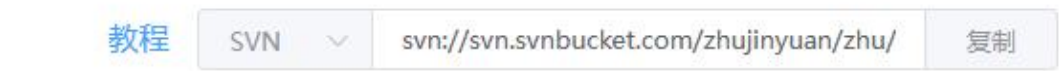


# 1. 拉項目

創建項目



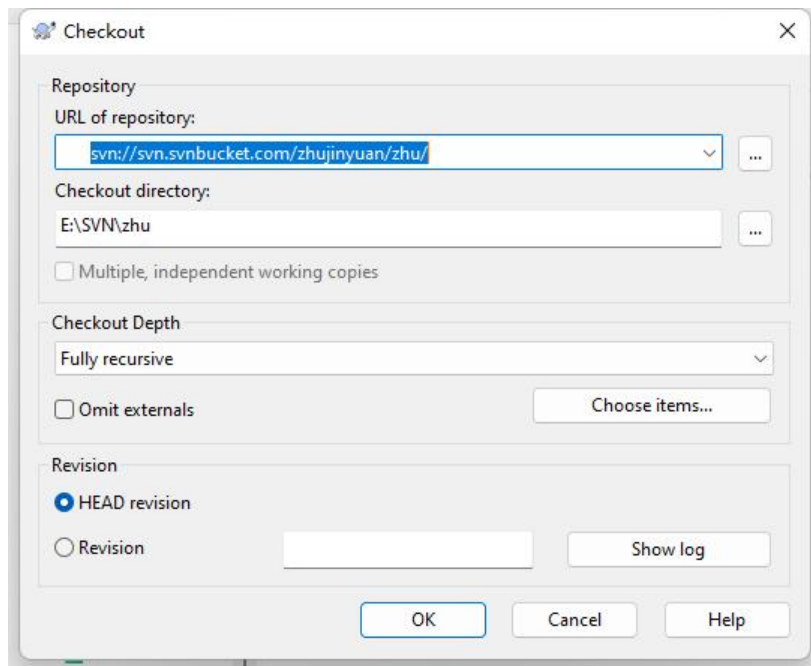
複製 url



右鍵點擊 SVN 的 checkout



選擇路徑



## 2. 更新

從 SVN 更新項目

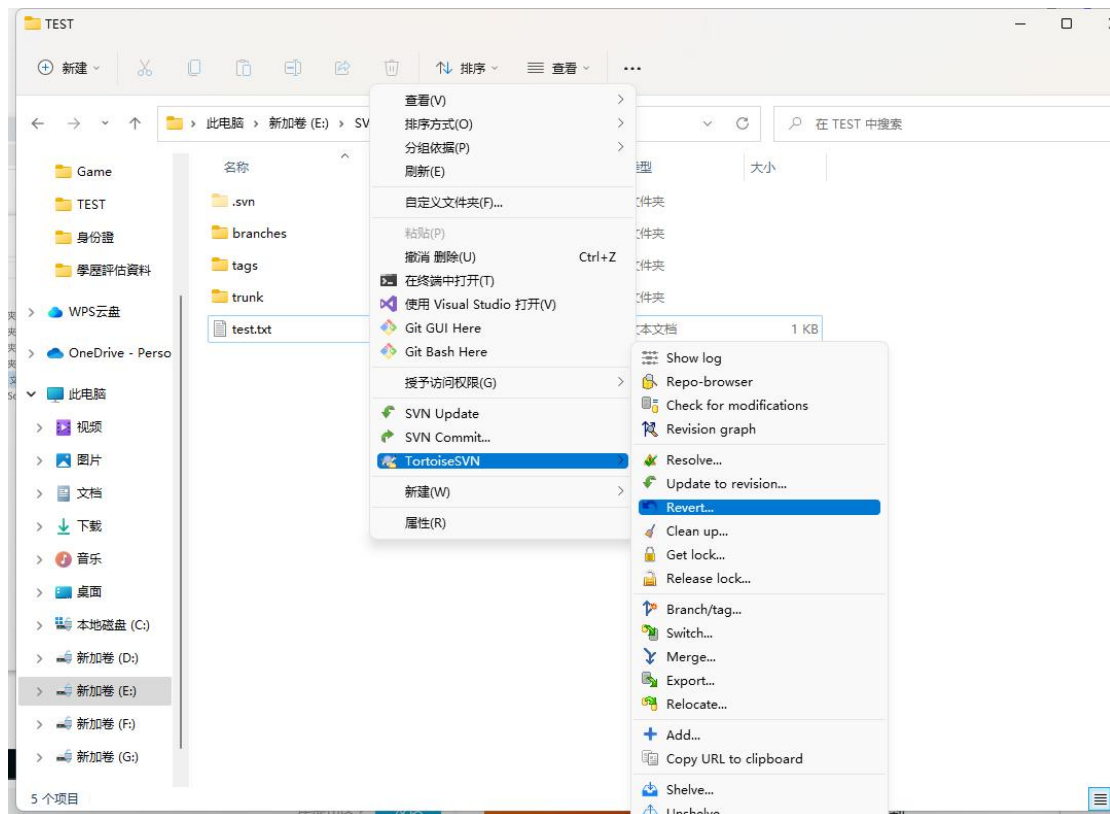
可以将旧版本的项目更新为最新版本



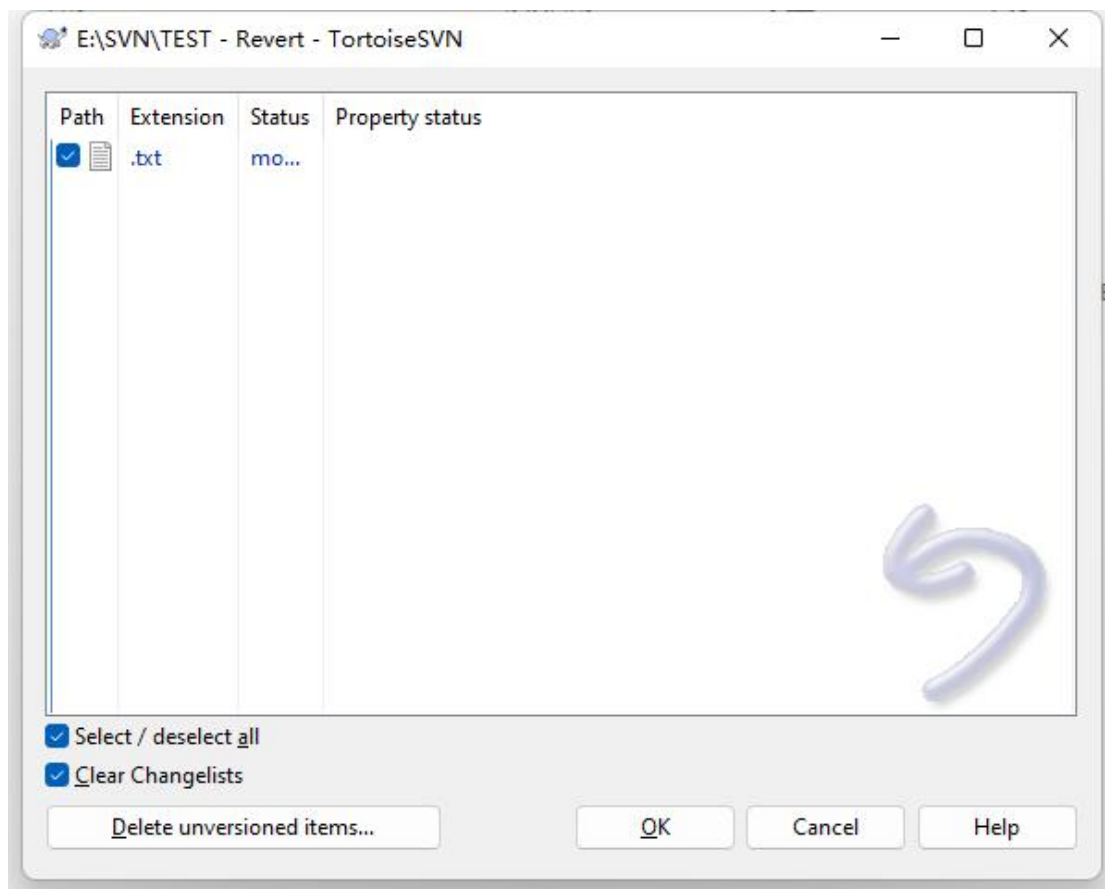
### 3. 上傳項目到 SVN



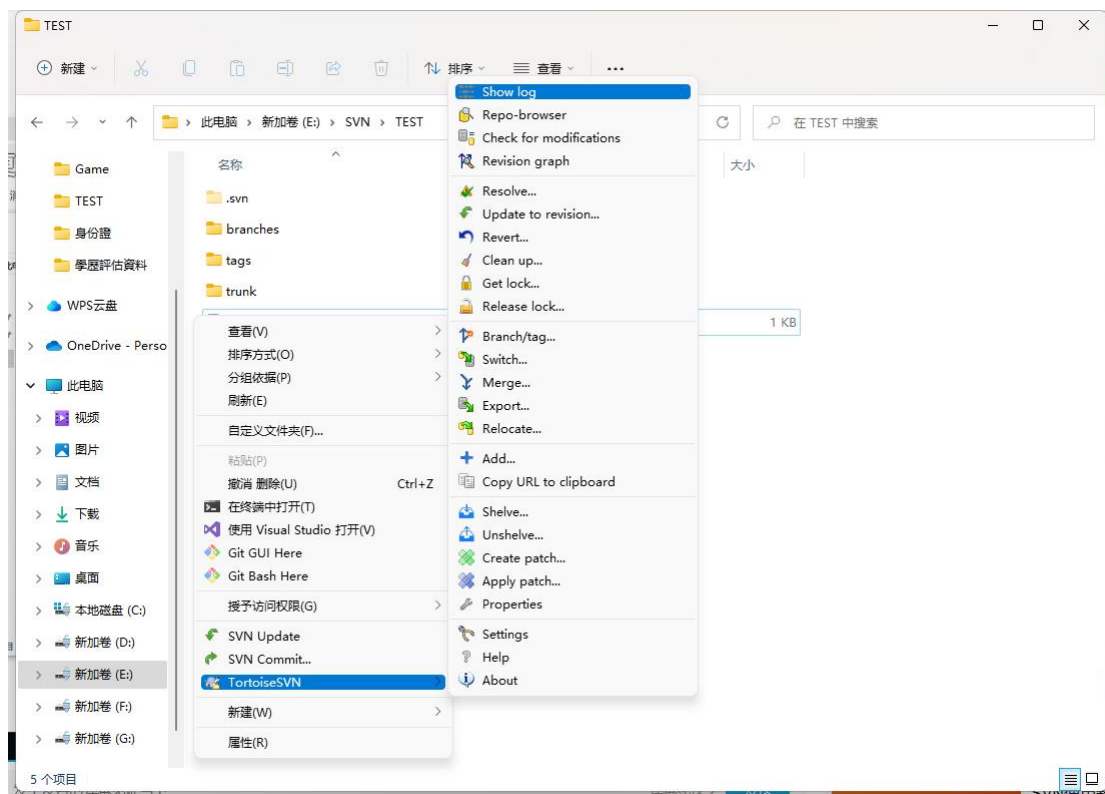
### 4. 當項目修改後，未上傳，想撤銷修改



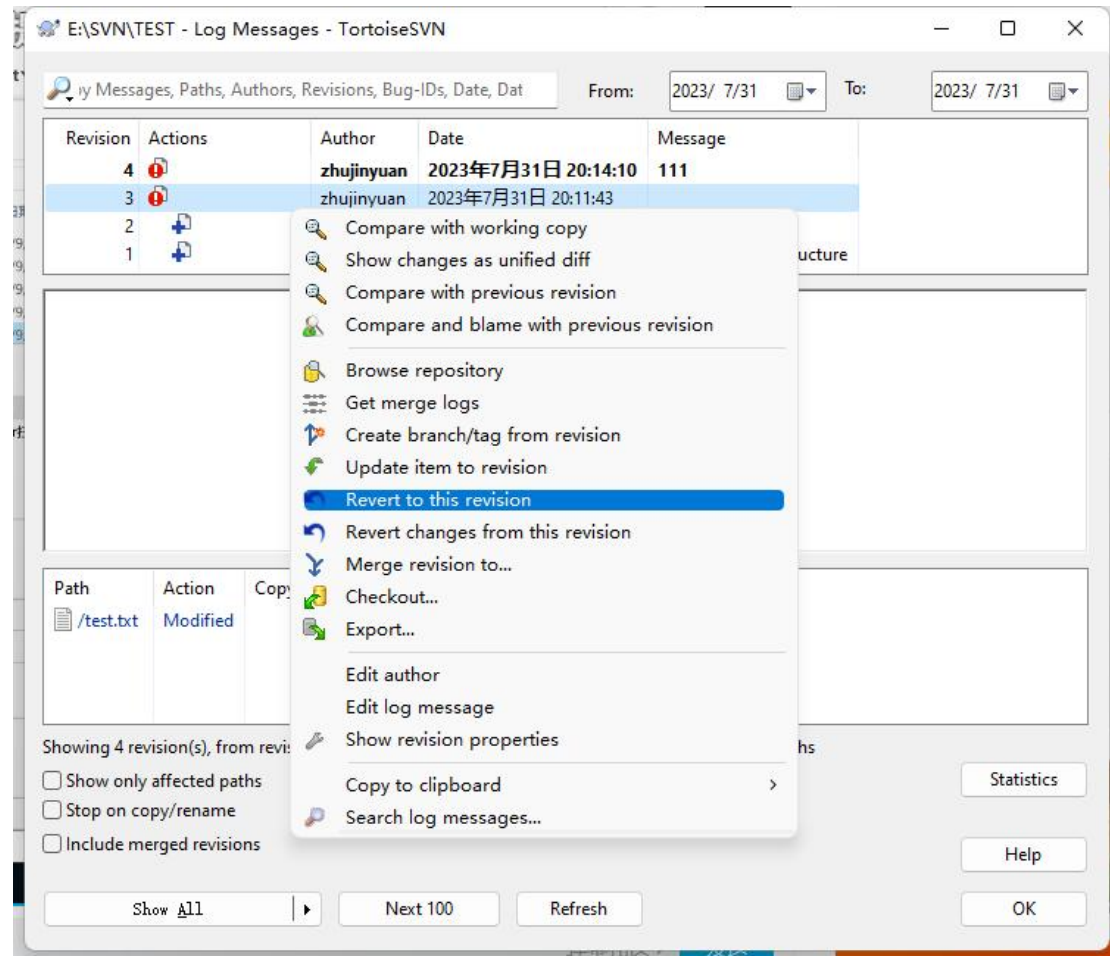
選擇撤銷的項目



## 5. 當項目修改後，已上傳，想撤銷修改

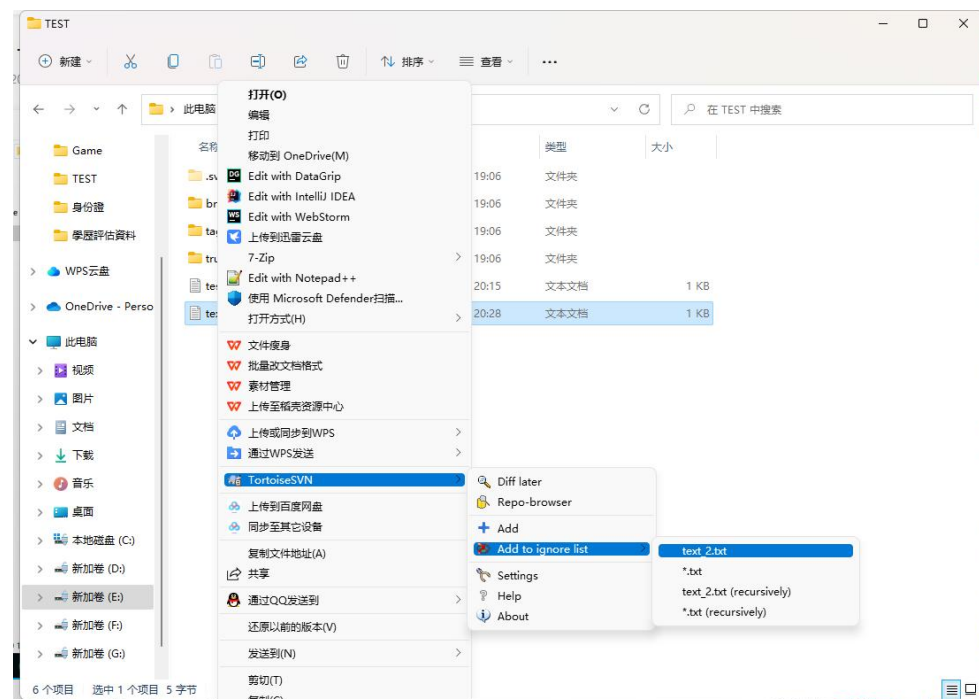


選擇想要復原的版本



## 6. 忽略文件上傳

選擇需要忽略的文件，再點擊上傳即可



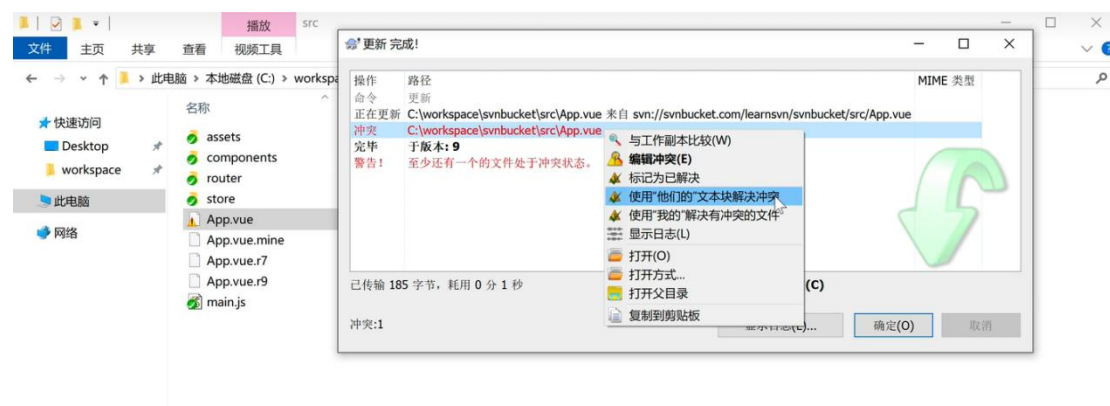
## 7. 解決衝突

什么情况容易发生冲突？  
多个人修改了同个文件的同一行  
无法进行合并的二进制文件

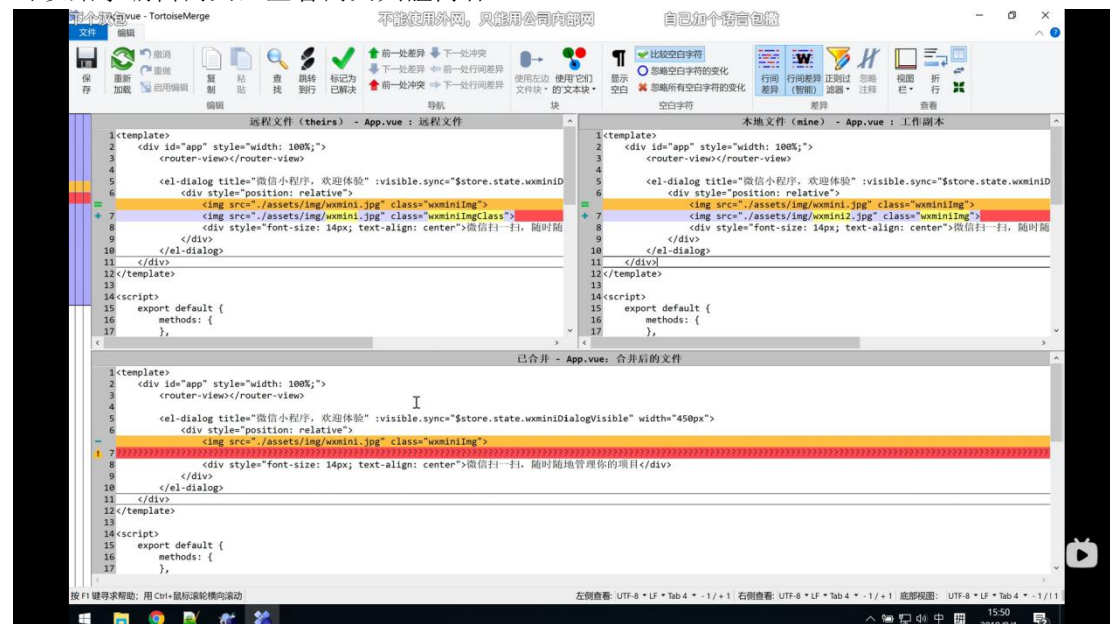
怎么避免冲突？  
经常update同步下他人的代码  
二进制文件不要多个人同时操作

冲突了怎么办？  
解决冲突很简单，不用怕

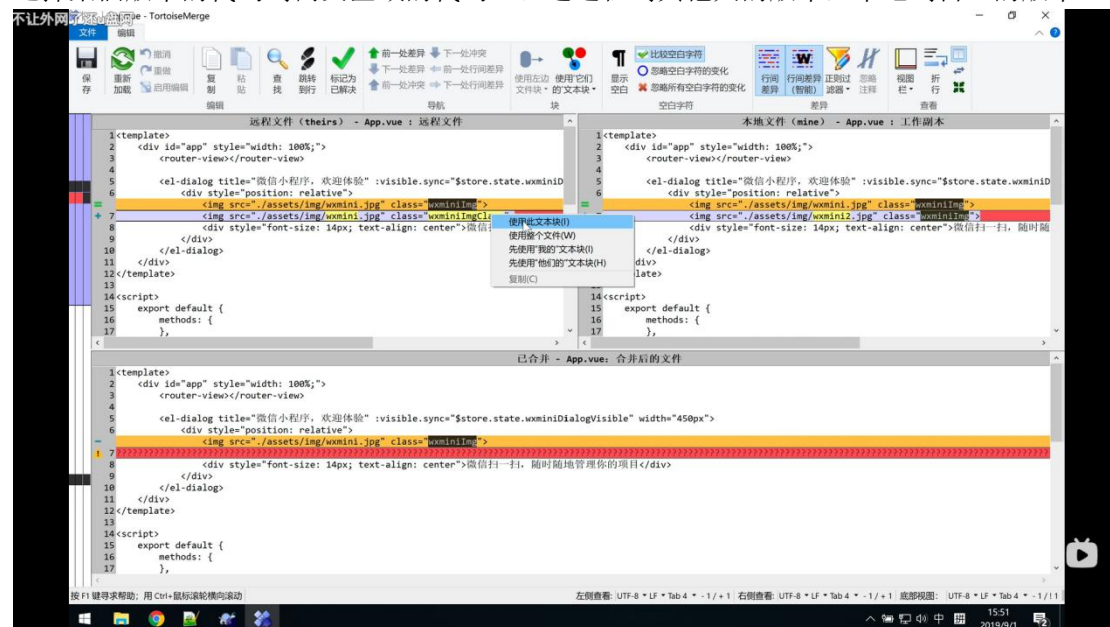
發生衝突時



可以點擊編輯衝突，查看衝突具體內容



選擇某個版本的代碼為衝突區域的代碼（左邊遠程為其他人的版本，本地為自己的版本）





## 8. 分支

什么时候需要开分支？

隔离线上版本和开发版本

大功能开发，不想影响到其他人，自己独立开个分支去开发

SVN经典目录结构：

trunk

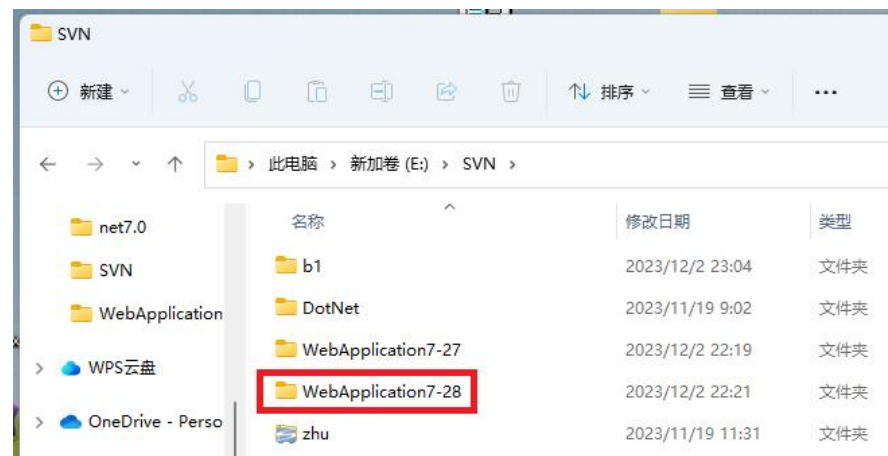
branches

tags

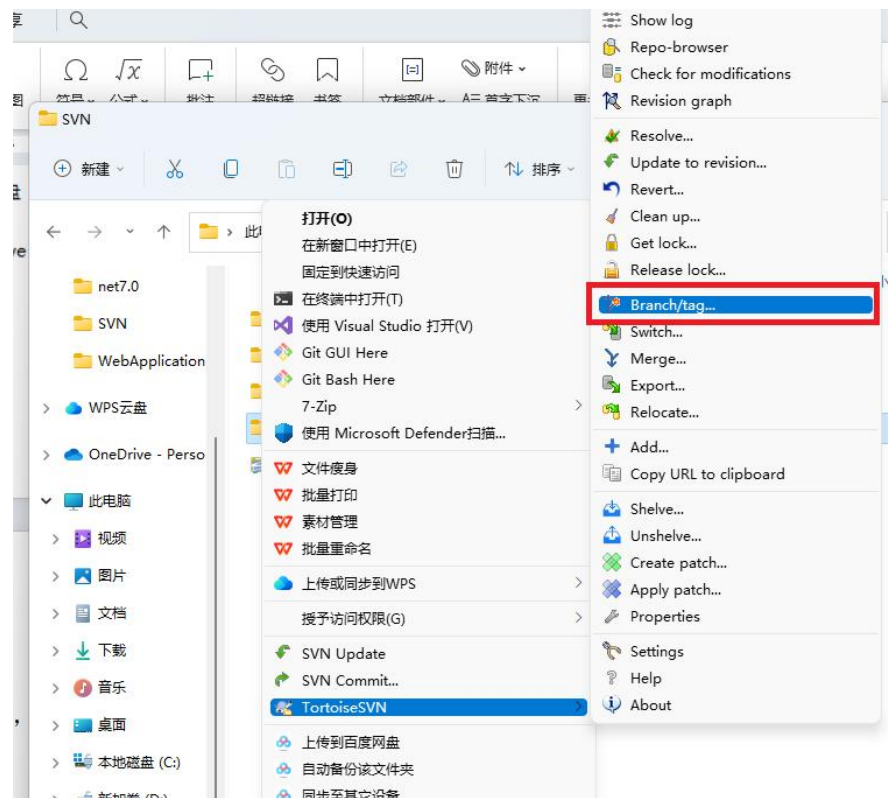
### 8.1 创建分支

Eg.

下载最新的版本到本地

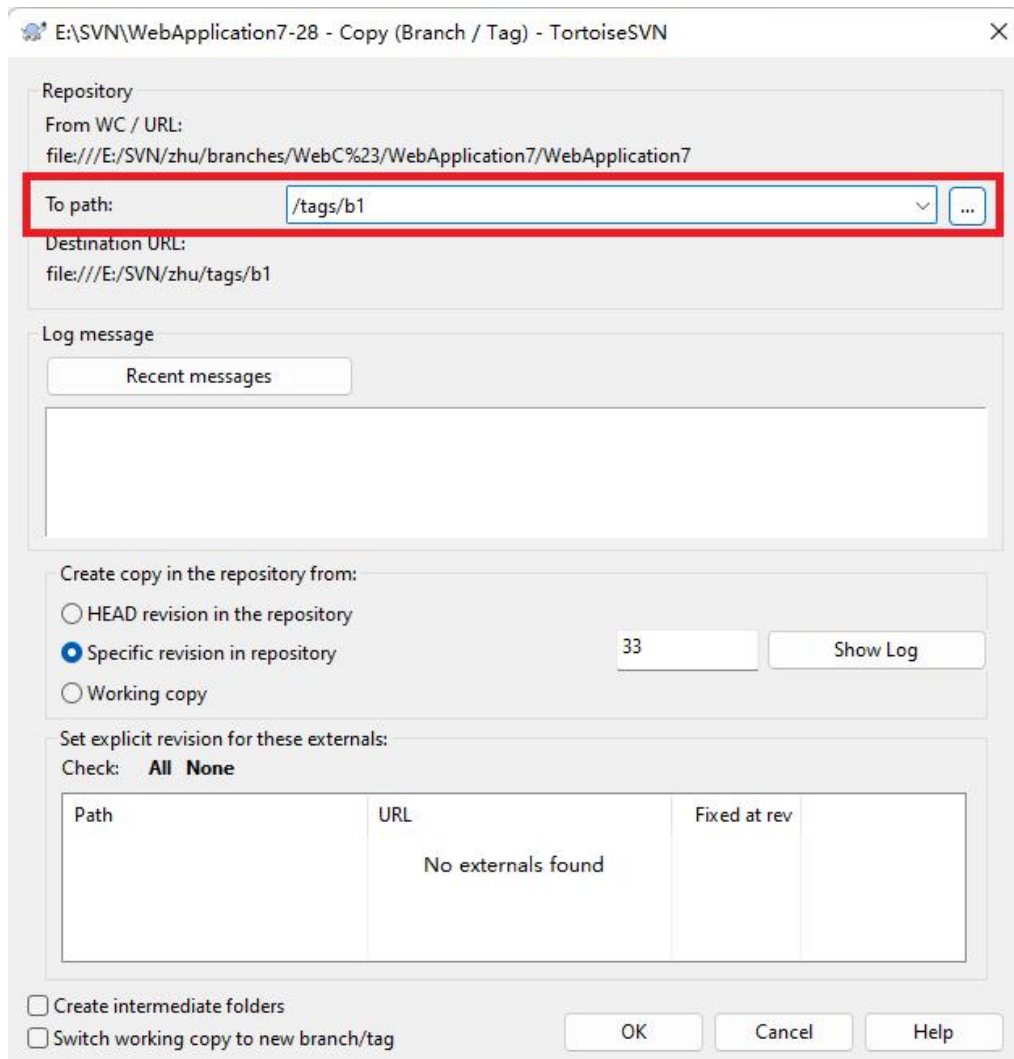


选择最新版本，选择 Branch/tag

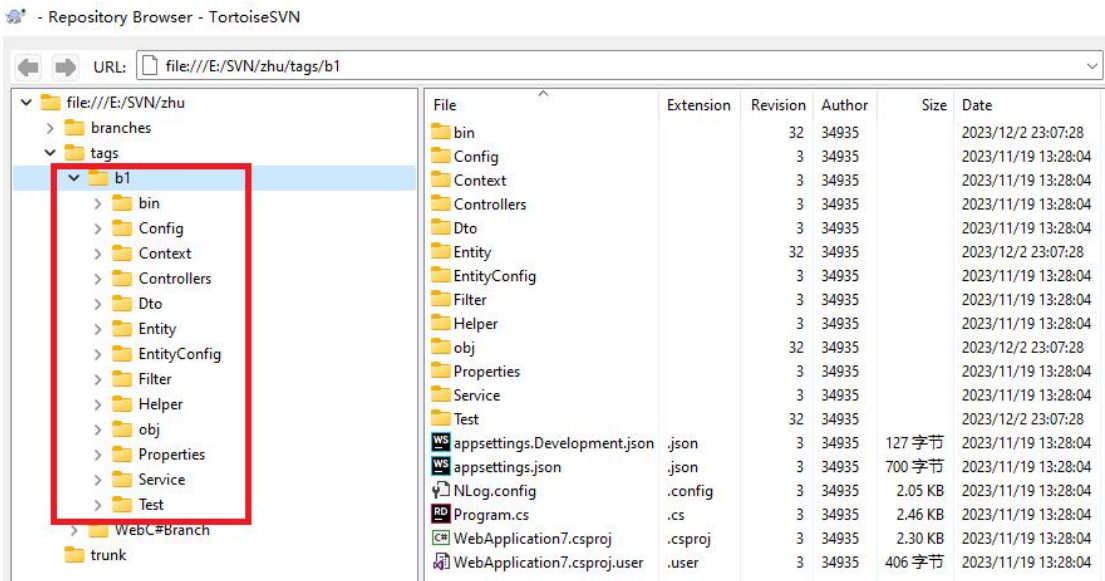




输入分支保存的路径



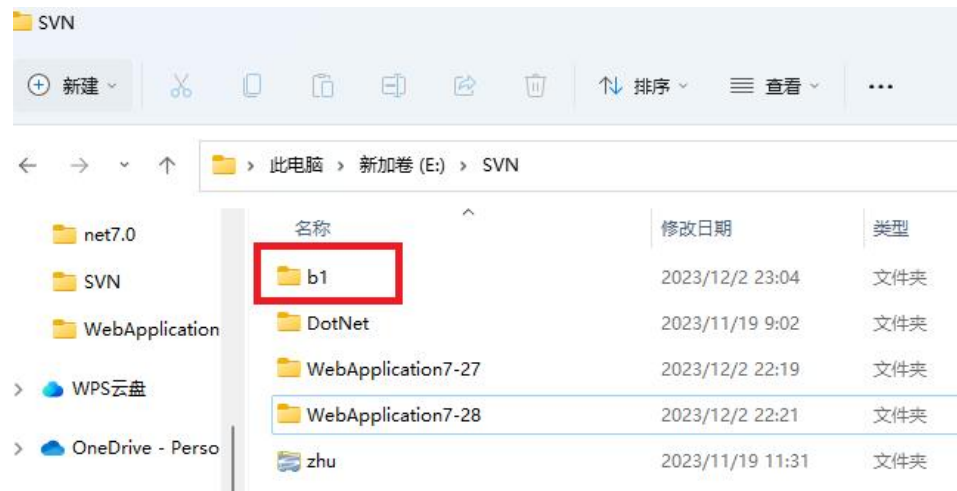
创建成功后，分支拥有和主干一样的文件



## 8.2 合并分支

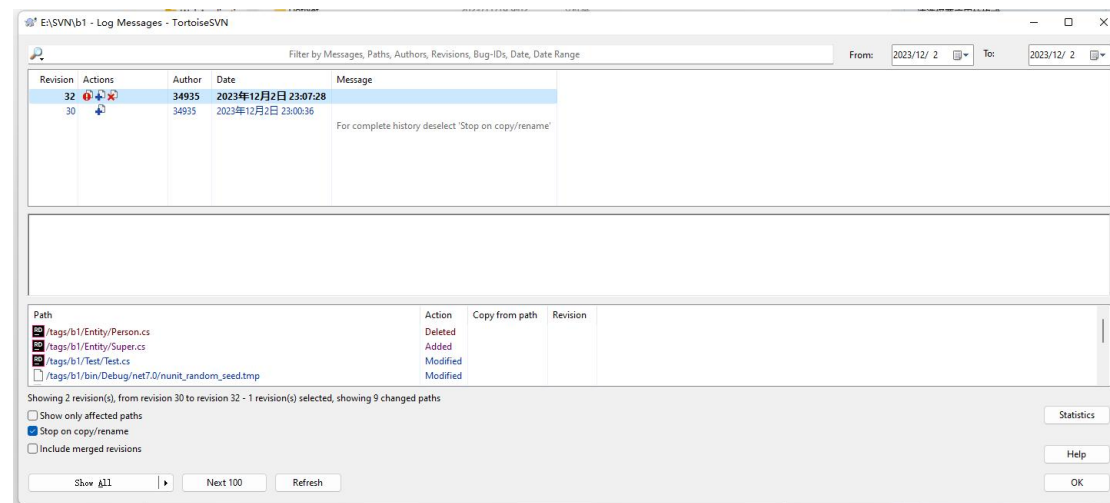
合并不要选 skip

用 checkout 下载分支

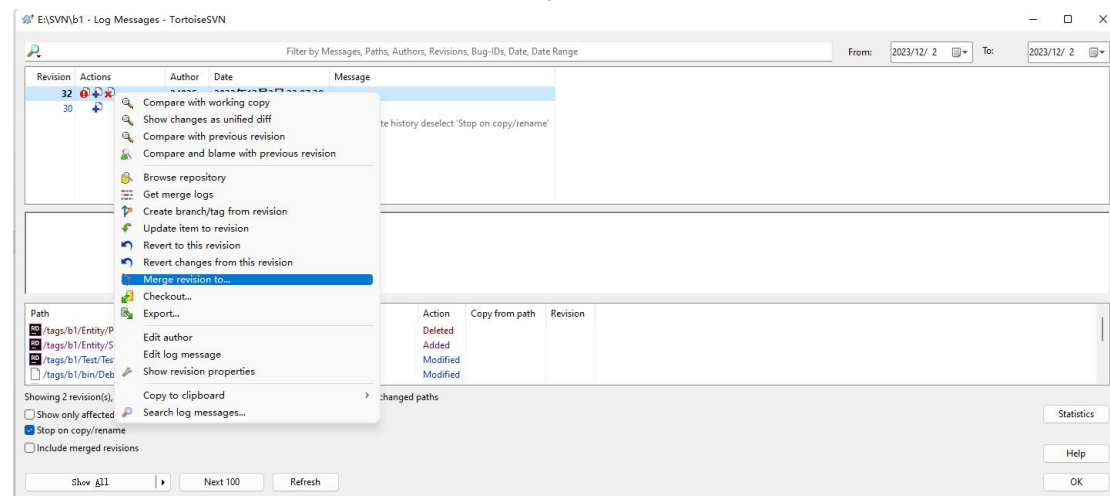


分支经历多个版本的开发后，需要和主干合并

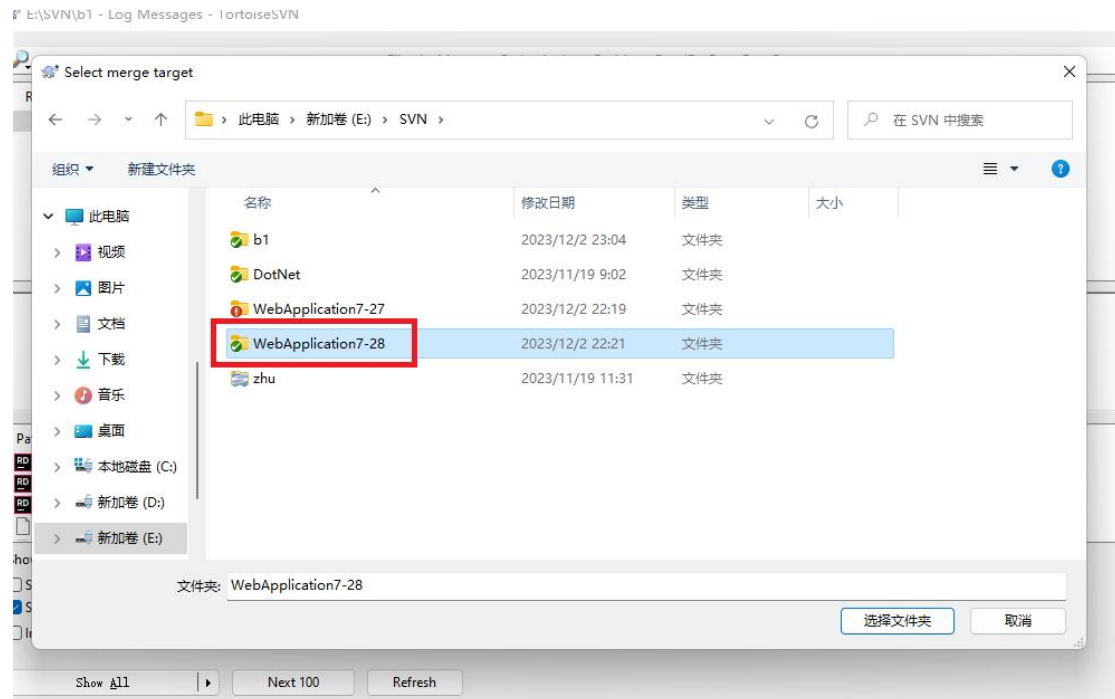
打开分支的 log



选择一个或多个需要合并的分支，选择 Merge revision to...



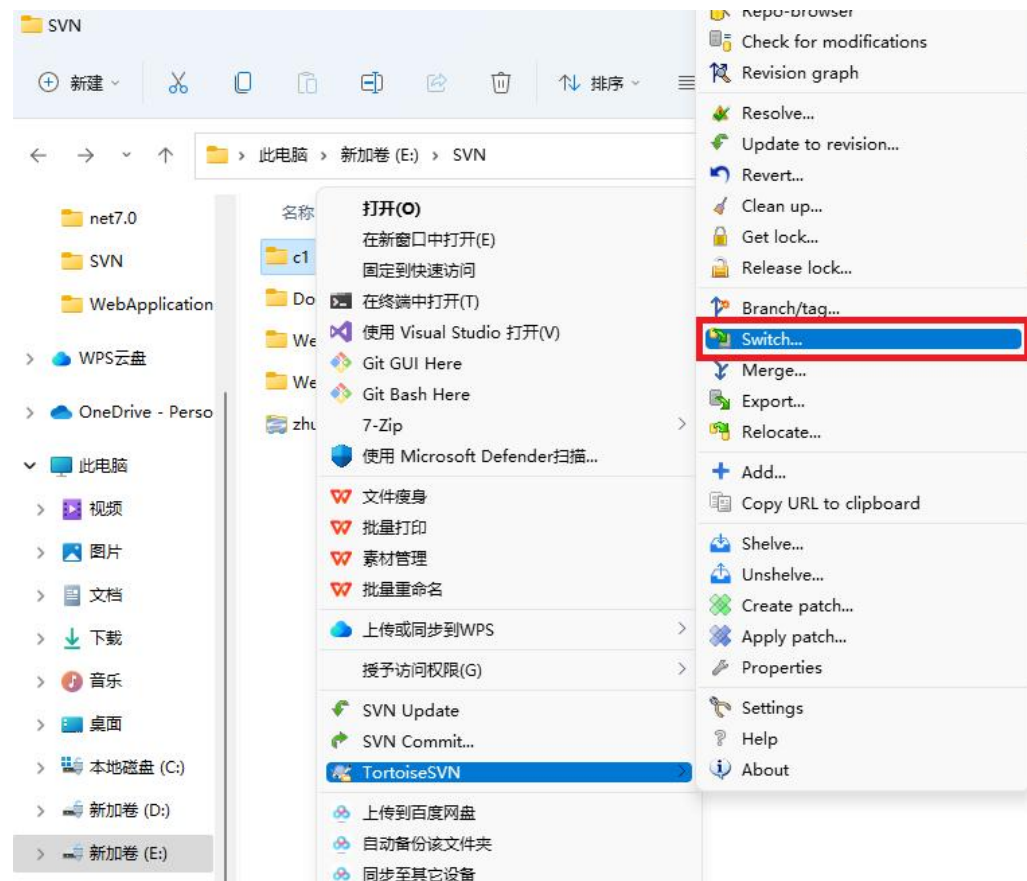
选择本地已经下载需要被合并的主干



合并后，将合并的主干版本提交到 SVN

### 8.3 切换分支

切换分支会让 A 分支的文件变为 B 分支的文件



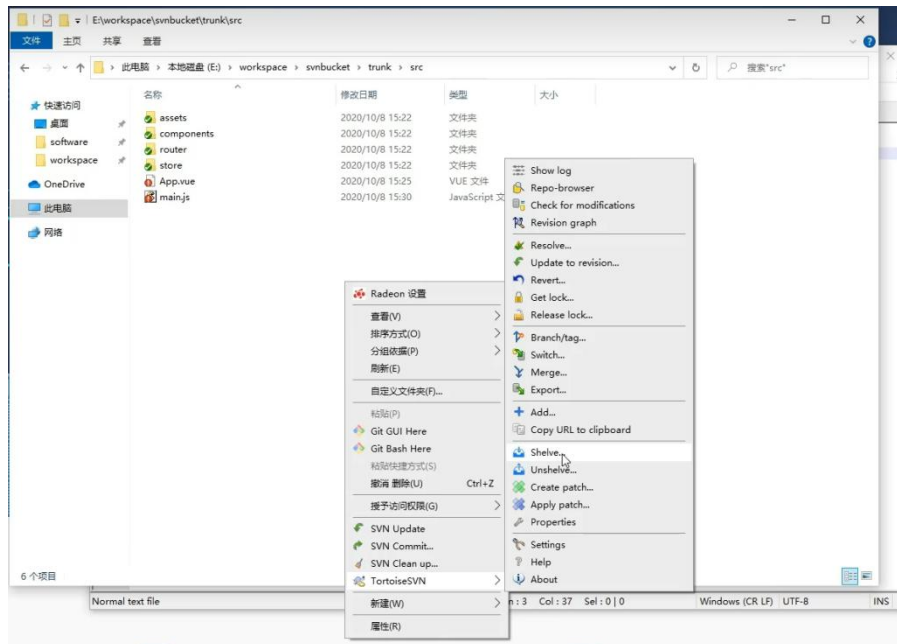
## 9. 暫存

有什么用？

1. 代码改了很多，突然需要紧急修复一个bug，但是代码还没写完，不能提交。
2. 代码重构了很多，突然需要发布新版本，但是代码还跑不起来，不能提交。

### 9.1 點擊 shelve (暫存)

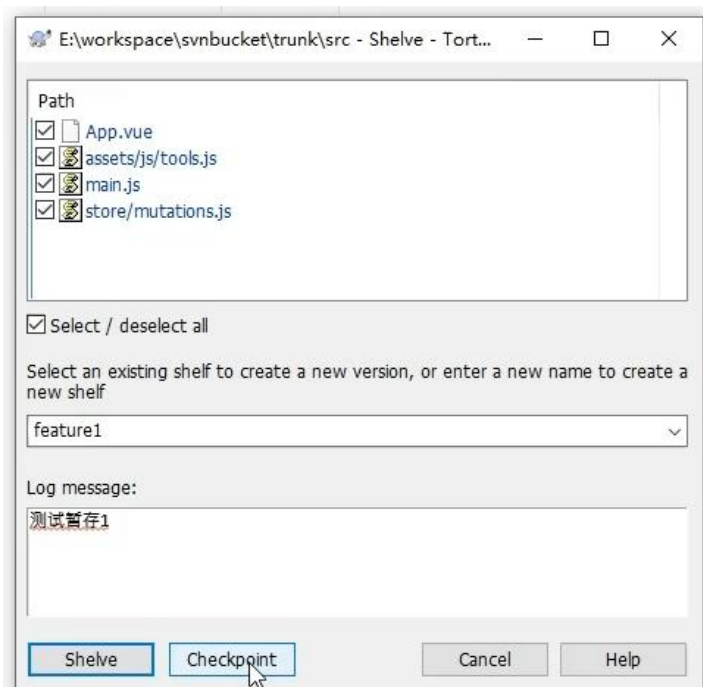
(unshelve 為取出暫存)



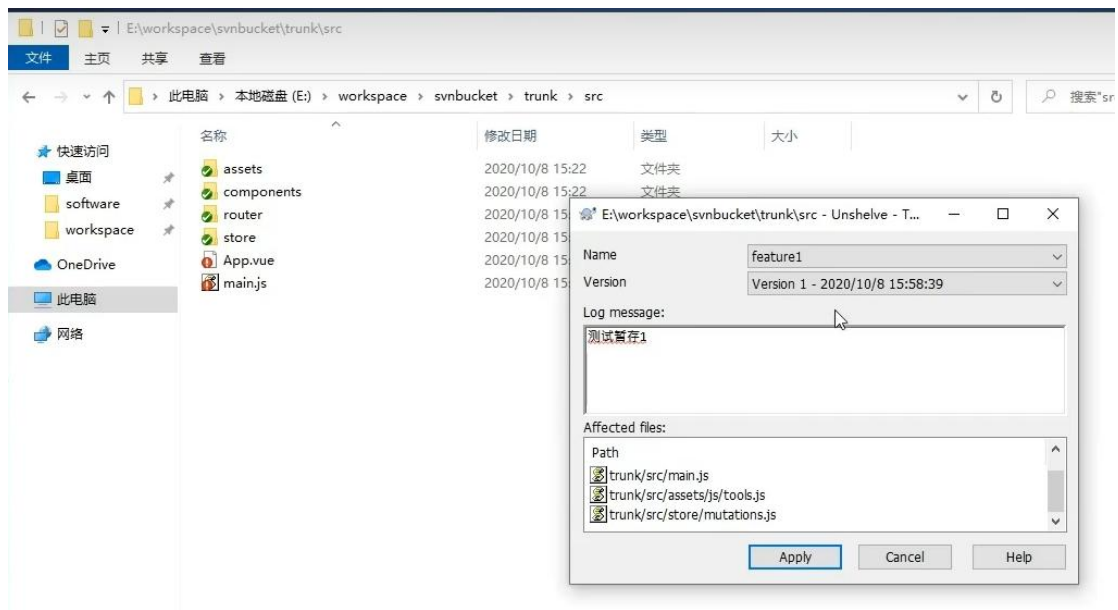
勾選暫存的文件

Shelve 按鈕會把本地修改撤銷掉，恢復成服務器的文件

Checkpoint 不會把本地修改撤銷掉，不會恢復成服務器的文件



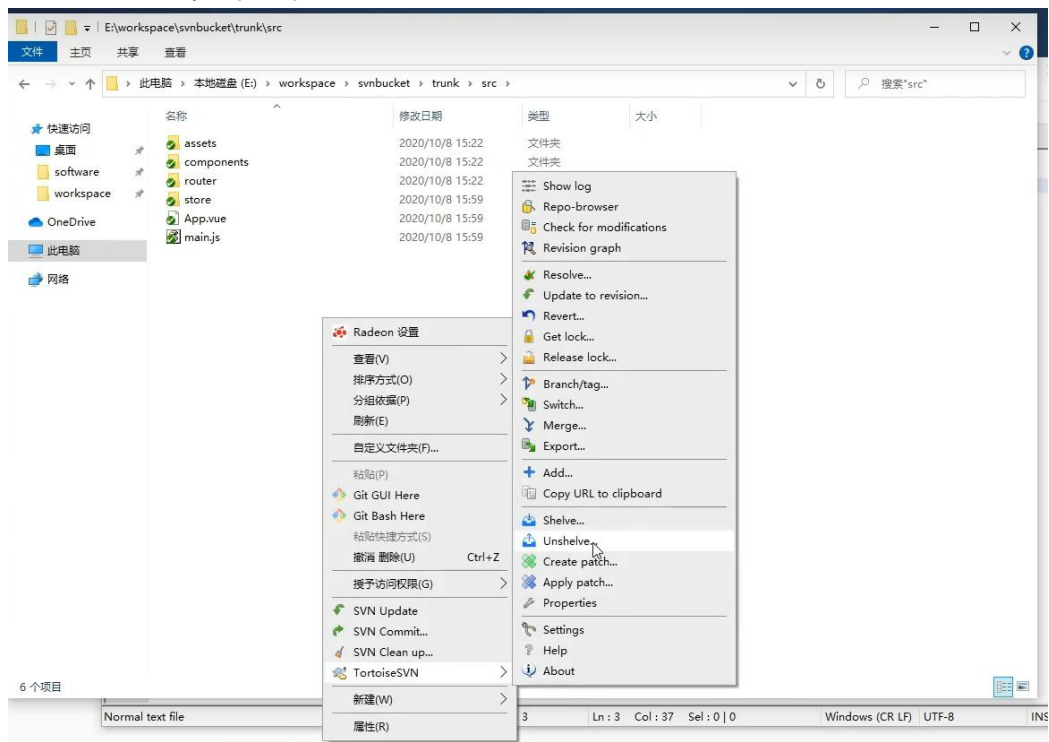
### 9.1.1 Checkprint 后

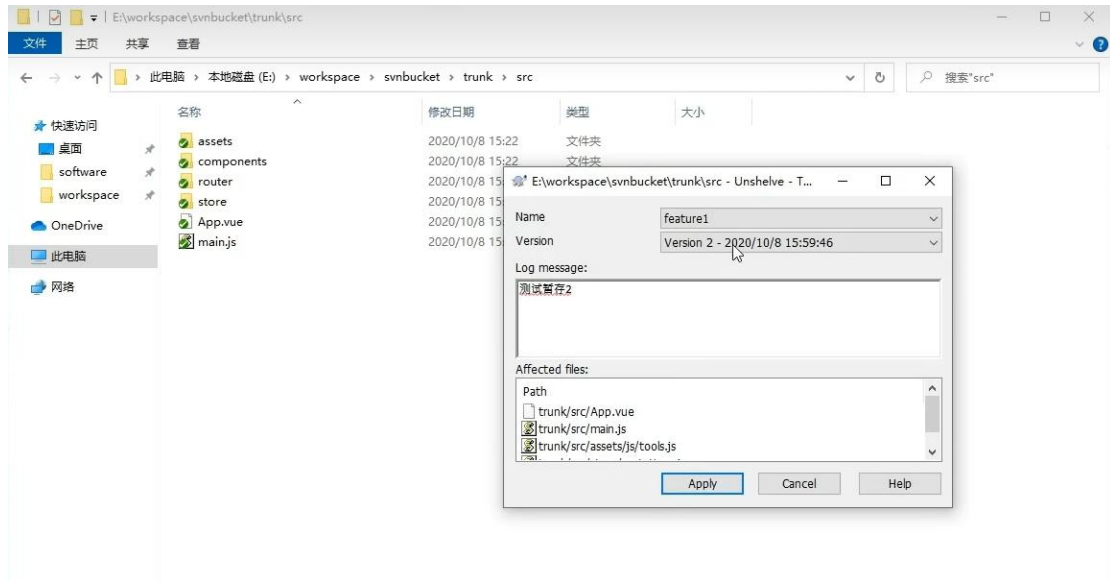


### 9.1.2 Shelve 后



## 9.2 unshelve 取出暂存







## 10. 合并

情况：

主干开发新功能，改了很多

分支是线上版本，修复了很多bug

两个分支的代码时间久了差异很大，无法直接分支合并或者指定提交记录合并代码

使用BeyondCompare

## 11. 版本回退

在讲解这三个选项的作用之前，我们还是先来假定一个使用情景，假设我们的项目文件一共有8个版本，它版本号分别是1, 2, 3, 4, 5, 6, 7, 8。

### Update item to this version

这个选项的作用是将文件版本更新到对应所选的版本（当然内容也修改到了相应的版本）。如果我们是在版本4这里点击“Update item to this version”，表示5~8版本所作的修改全部作废，这个文件的历史回退到了版本4那个时代，但是需要注意的是，此时文件的版本是4，并不是最新的。我们知道SVN工具中如果文件不是最新版本就无法上传，所以说这个功能只是用来暂时还原一下版本，来查询某个问题的，不能将还原后的文件上传。

### Revert to this version

这个选项的作用是将文件的内容更新到对应的版本，版本号没有发生变化。如果我们是在版本4这里点击“Revert to this version”，表示5~8版本所作的修改全部被还原，文件和版本4的文件一模一样，但需要注意的是这项操作相当于我们把版本4这个文件拷贝了一份赋值给了当前目录下的文件，此时的文件版本还是8，并且是可以提交的，提交以后文件的版本变成了9，增加了一个新的版本，虽然这个版本和版本4的内容是一样的。

### Revert changes from this version

这个选项的作用是将对应版本的修改还原，文件的版本号不发生变化，相当于在当前本版本上剔除某些版本所作的改变。如果我们是在版本4这里点击“Revert changes from this version”，表示版本4所作的修改被抹杀了，只剩下除版本4以外的7个修改了，但是此时文件是可以上传的，并且会生成新的版本9，只是版本9只包括除版本4以外的7次修改。这个选项是可以选择多个版本的，如果我们选择4,5,6,7这四个版本点击“Revert changes from this revision”，那么这几次修改都会被抹杀。如果我们选择5,6,7,8这四个版本点击“Revert changes from this revision”，表示取消这几个版本的修改，实际上和在版本4这里点击“Revert to this version”的作用是一样的。

## 12. 多仓库管理项目

不同仓库对应不同项目，适用于同时开发多个项目

### 九、配置多仓库与权限控制

#### 1、配置多仓库

在实际项目开发中，我们可能会同时开发多个项目，那么我们如何进行多项目监管呢？

通过 svnserve 进行仓库监管，但是监管指令只能监管某一个文件夹，而不能同时监管多个仓库。

答：可以通过监管 WebApp 总目录来达到监管所有仓库的目的。

svnserve -d(后台运行) -r(监管目录) WebApp (项目总目录)

① 打开 DOS 窗口，输入如下指令

```
C:\Users\Administrator>svnserve -d -r D:/svn/WebApp
```

svn://localhost 或 ip 地址来访问 D:/svn/WebApp 目录

如果需要访问 Shop 项目、Wechat 项目

Shop 项目： svn://localhost/Shop

Wechat 项目： svn://localhost/Wechat

## 13. 权限控制

### 2、权限控制

如果要使用权限控制有一个前提：必须首先开启权限功能。

在每一个仓库中都有一个 `conf` 文件夹，里面有三个文件。

`authz` 文件：授权文件。

告诉哪些用户具有哪些权限。

。

`passwd` 文件：认证文件。

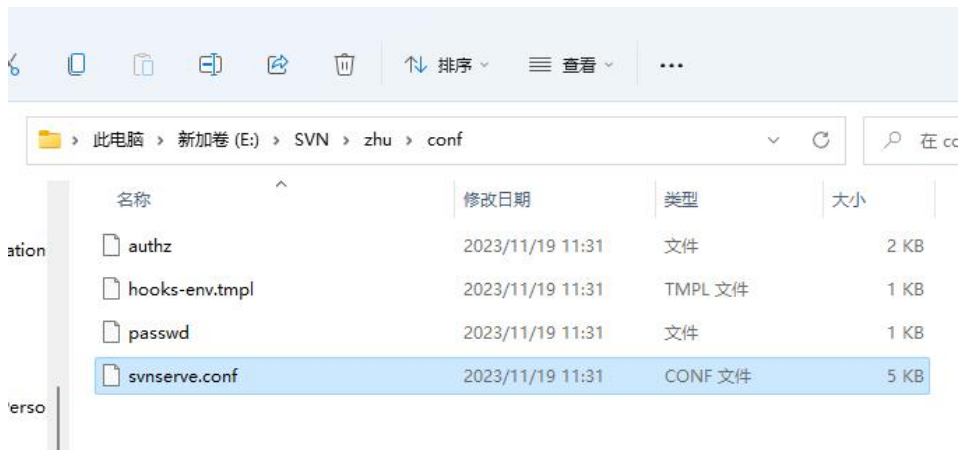
标识当前 `svn` 系统中某个仓库具有哪些用户以及相应的密码。

。

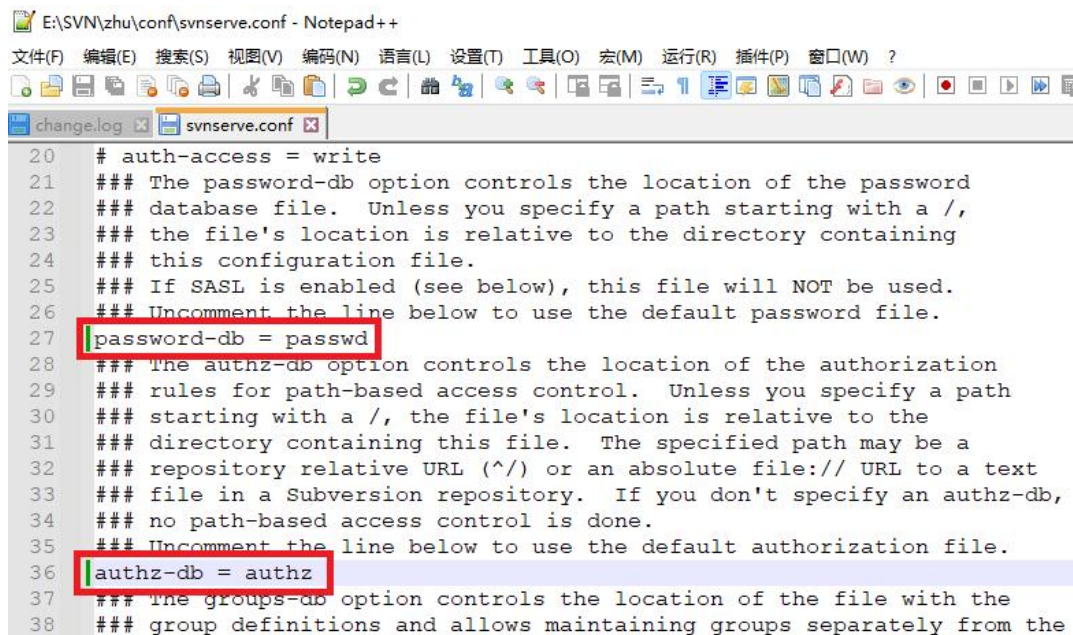
默认情况下，以上两个文件都是禁用的，如需要使用，首先要开启以上两个文件。

`svnserve.conf` 配置文件。

每个仓库都有各自的 `Svnserve.conf`，如以下仓库。

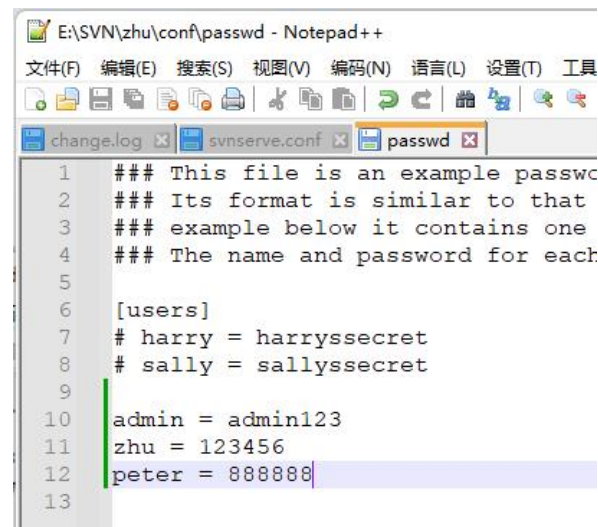


修改 `Svnserve.conf`



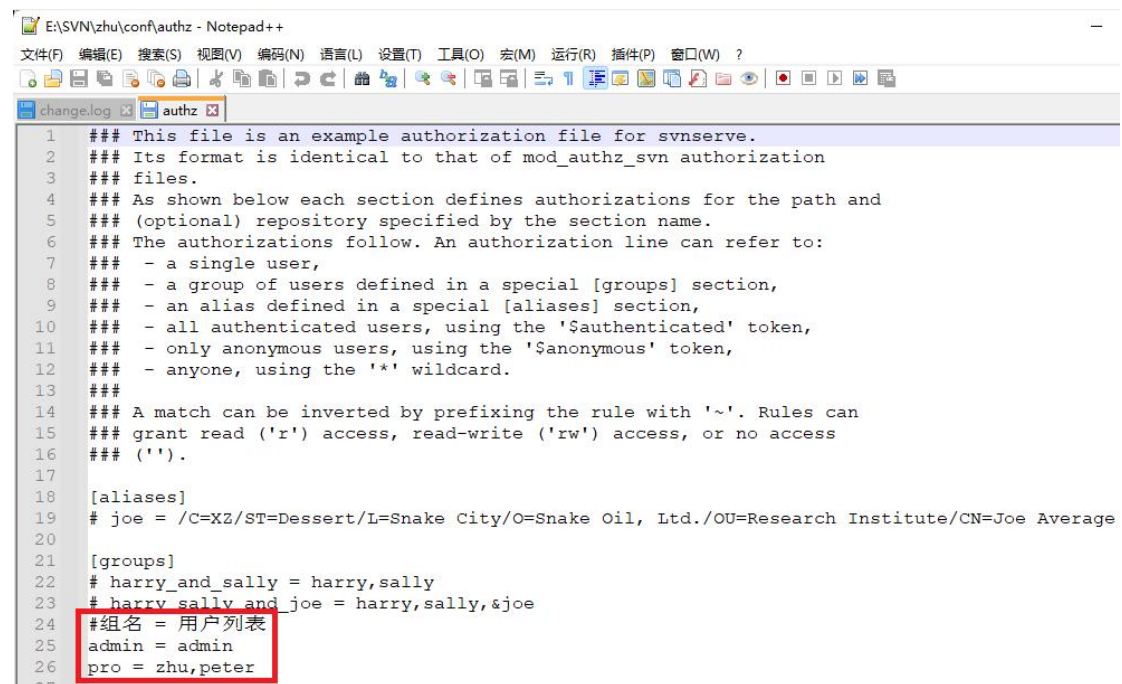
## 修改 password

左边为账号，右边为密码



```
E:\SVN\zhu\conf\passwd - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具
change.log x svnservice.conf x passwd x
1  ### This file is an example passwd
2  ### Its format is similar to that
3  ### example below it contains one
4  ### The name and password for each
5
6  [users]
7  # harry = harryssecret
8  # sally = sallyssecret
9
10 admin = admin123
11 zhu = 123456
12 peter = 888888
13
```

## 修改 authz



```
E:\SVN\zhu\conf\authz - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
change.log x authz x
1  ### This file is an example authorization file for svnserve.
2  ### Its format is identical to that of mod_authz_svn authorization
3  ### files.
4  ### As shown below each section defines authorizations for the path and
5  ### (optional) repository specified by the section name.
6  ### The authorizations follow. An authorization line can refer to:
7  ### - a single user,
8  ### - a group of users defined in a special [groups] section,
9  ### - an alias defined in a special [aliases] section,
10 ### - all authenticated users, using the '$authenticated' token,
11 ### - only anonymous users, using the '$anonymous' token,
12 ### - anyone, using the '*' wildcard.
13 ###
14 ### A match can be inverted by prefixing the rule with '~'. Rules can
15 ### grant read ('r') access, read-write ('rw') access, or no access
16 ### ('').
17
18 [aliases]
19 # joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil, Ltd./OU=Research Institute/CN=Joe Average
20
21 [groups]
22 # harry_and_sally = harry,sally
23 # harry_sally_and_joe = harry,sally,&joe
24 #组名 = 用户列表
25 admin = admin
26 pro = zhu,peter
27
```

