**Sourcetree使用手册**

|  |  |  |
| --- | --- | --- |
| 文件状态  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文档编号 |  |
| 项目编号 |  |
| 项目经理 |  |
| 当前版本 | V1.0 |
| 完成日期 | 2018年10月8日 |

sunline_logo

深圳市长亮科技股份有限公司

Shenzhen Sunline Technology Co., Ltd.

**文档修订记录**

| 修订日期 | 版本 | 变更描述 | 修订方法 | 修订说明 | 修订人 |
| --- | --- | --- | --- | --- | --- |
| 2018/10/8 | V1.0 | 创建文档 | 新增 | 初稿 | 张春阳 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

修订方法：新增/修改/删除

目录

[1 引言 4](#_Toc526771883)

[1.1 编写目的 4](#_Toc526771884)

[1.2 术语表 4](#_Toc526771885)

[1.3 预期读者 4](#_Toc526771886)

[1.4 参考文献 4](#_Toc526771887)

[2 软件背景 4](#_Toc526771888)

[3 软件简介 5](#_Toc526771889)

[4 使用配置和界面 5](#_Toc526771890)

[4.1 交易控制器表(ksys\_jykzhq) 6](#_Toc526771891)

[4.2 任务执行域表(ksys\_rwzhxy) 7](#_Toc526771892)

[4.3 定时任务调度信息表(ksys\_dsrwdd) 8](#_Toc526771893)

[4.4 运行许可条件表(ksys\_yxxktj) 9](#_Toc526771894)

[4.5 组件实例属性表(ksys\_zjslsx) 10](#_Toc526771895)

[4.6 Setting配置文件 10](#_Toc526771896)

[5 常见FAQ 11](#_Toc526771897)

# 引言

## 编写目的

本文档为sourcetree使用提供支持，经供参考。

## 术语表

|  |  |
| --- | --- |
| 术语 | 解 释 |
|  |  |

## 预期读者

项目开发人员；

项目管理部领导

## 参考文献

| 文档名称 | 文档标识 | 作者 | 备注 |
| --- | --- | --- | --- |
|  |  |  |  |

# 软件背景

* **Git：**

(/gɪt/)是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。

* **版权公司：**

**Atlassian**是一家澳大利亚软件公司，其产品面向软件开发人员，项目管理者和内容管理。它以其问题跟踪应用程序Jira及其团队协作和wiki产品Confluence而闻名。Atlassian为超过120,000名客户提供服务。

(参考wikipedia**:Atlassian Corporation Plc** (/ətˈlæsiən/) is an Australian enterprise software company that develops products for software developers, project managers, and content management.It is best known for its issue tracking application, Jira, and its team collaboration and wiki product, Confluence.Atlassian serves over 120,000 customers)

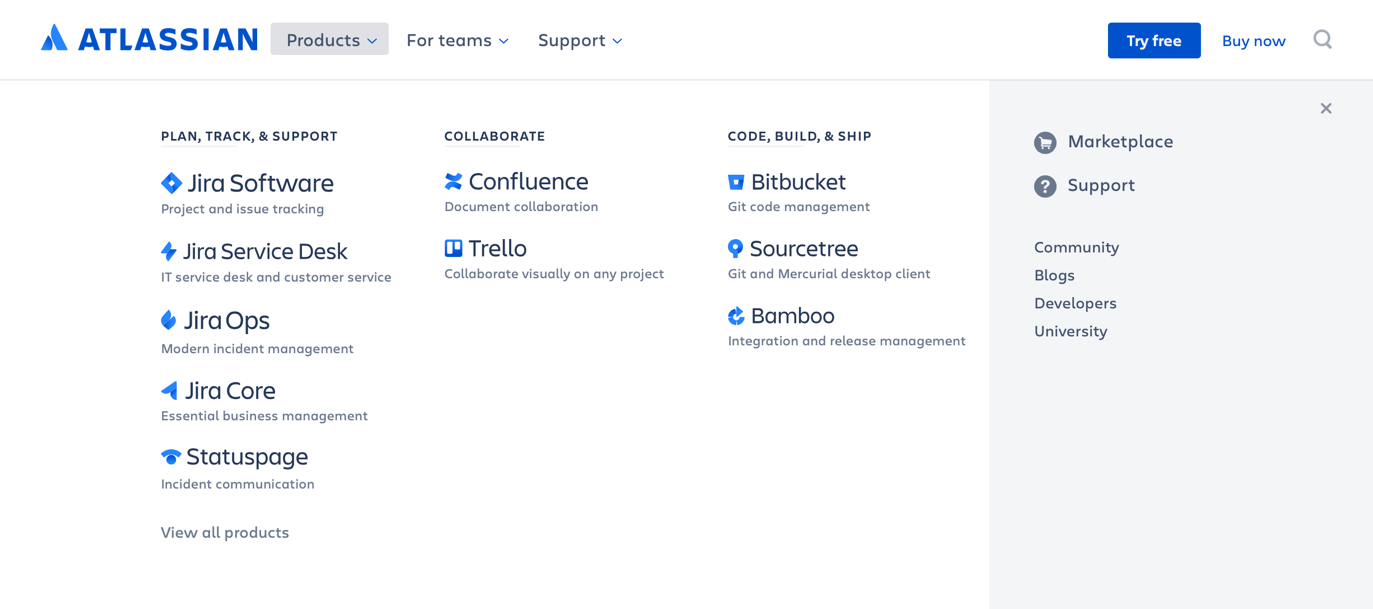


图1 ATLASSIAN产品

# 软件简介

SourceTree 是 Windows 和Mac OS X 下免费的 Git 和 Hg 客户端管理工具，同时也是Mercurial和Subversion版本控制系统工具。支持创建、克隆、提交、push、pull 和合并等操作。

SourceTree拥有一个精美简洁的界面，大大简化了开发者与代码库之间的Git操作方式，这对于那些不熟悉Git命令的开发者来说非常实用。

**SourceTree拥有完整的Git功能：**

* 通过一个简单的用户界面即可使用所有的Git命令
* 通过一次单击，即可管理所有的Git库，无论是托管的还是本地的
* 通过一次单击，即可进行commit、push、pull、merge等操作
* 一些先进的功能，如补丁处理、rebase、shelve、cherry picking等
* 可以连接到你托管在Bitbucket、Stash、Microsoft TFS或GitHub中的代码库

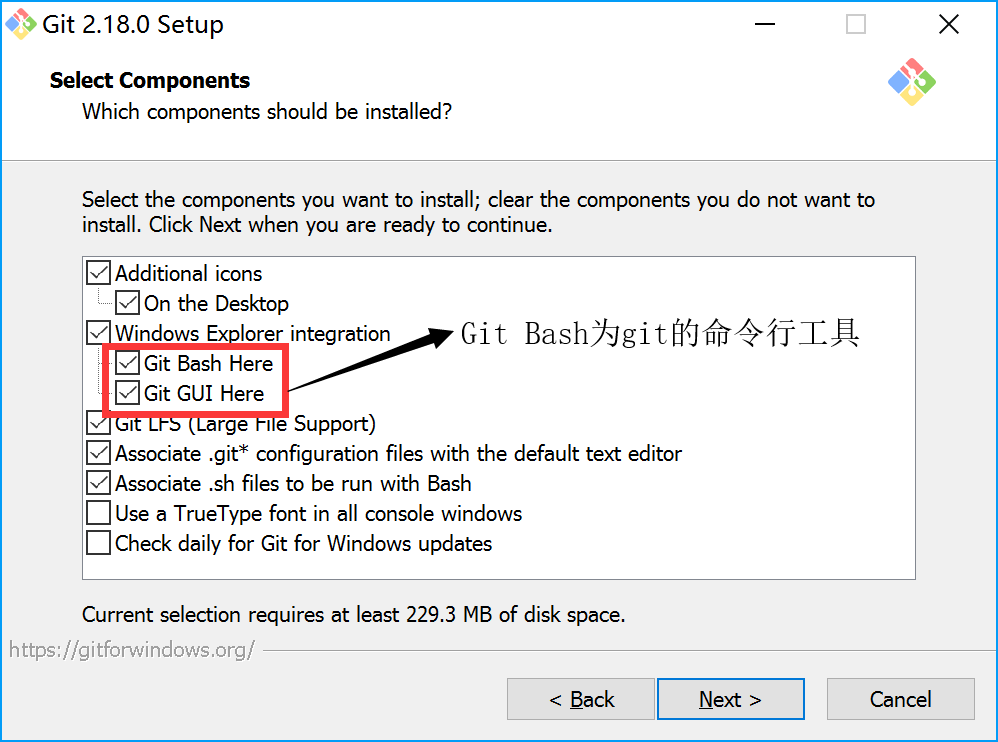
处理。

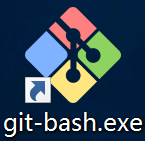
# 安装和配置

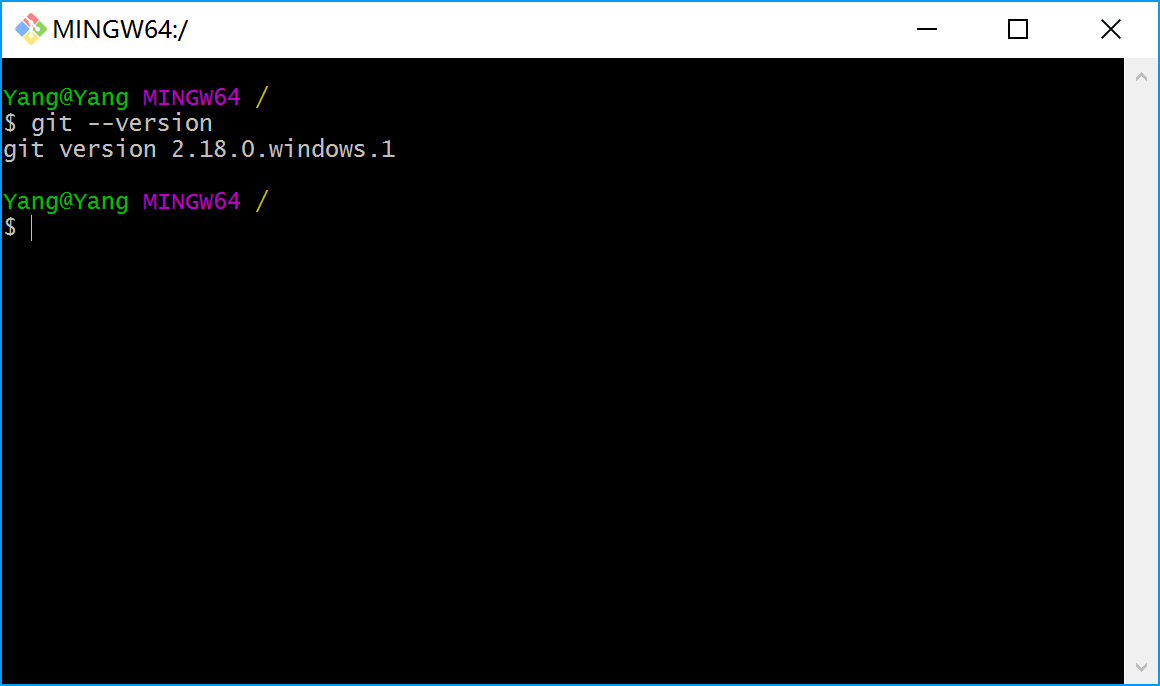
## Git安装配置

* + **须知：**使用Sourcetree需要先安装Git版本控制工具。已经安装所需版本Git或者有最新Sourcetree版本可跳过此步骤 (最新版的Sourcetree内置了git)
  + **下载：**登录官网下载安装git(<https://git-scm.com/>)。

图2 Git官网

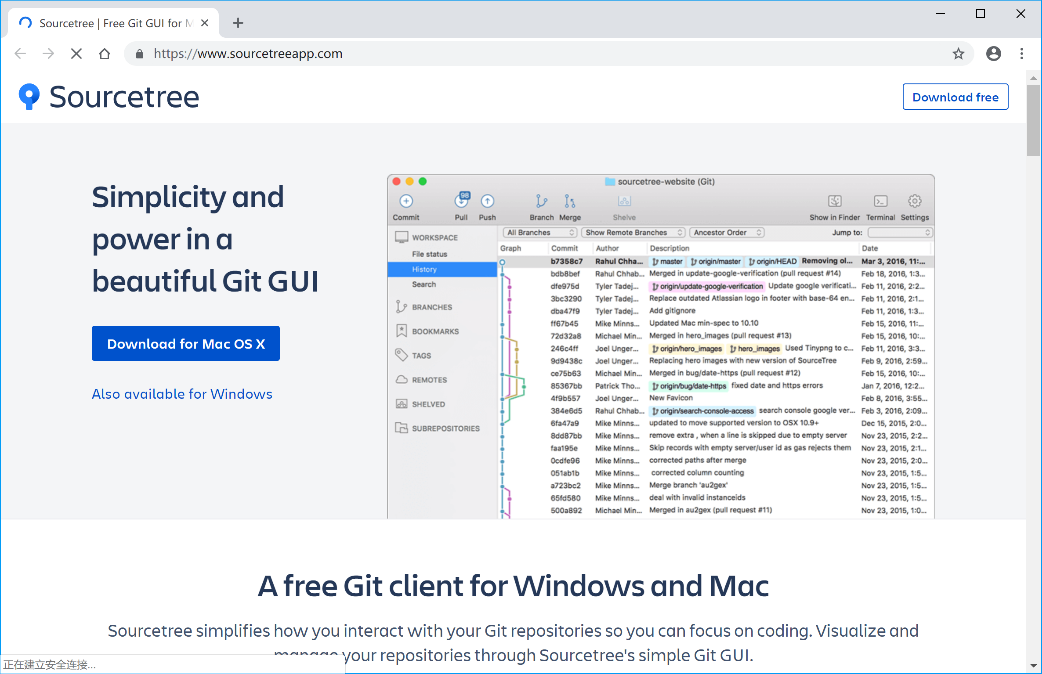
* + **安装：** 默认选项安装即可。GitBash为git 的命令行工具，熟练使用git命令可以更快捷的使用git。但命令繁多，且对初学者并不太友好，故才需要图形化的Git管理工具。

* + **测试：**安装完成，打开命令行工具测试。键入git –version 命令，返回版本即安装成功。

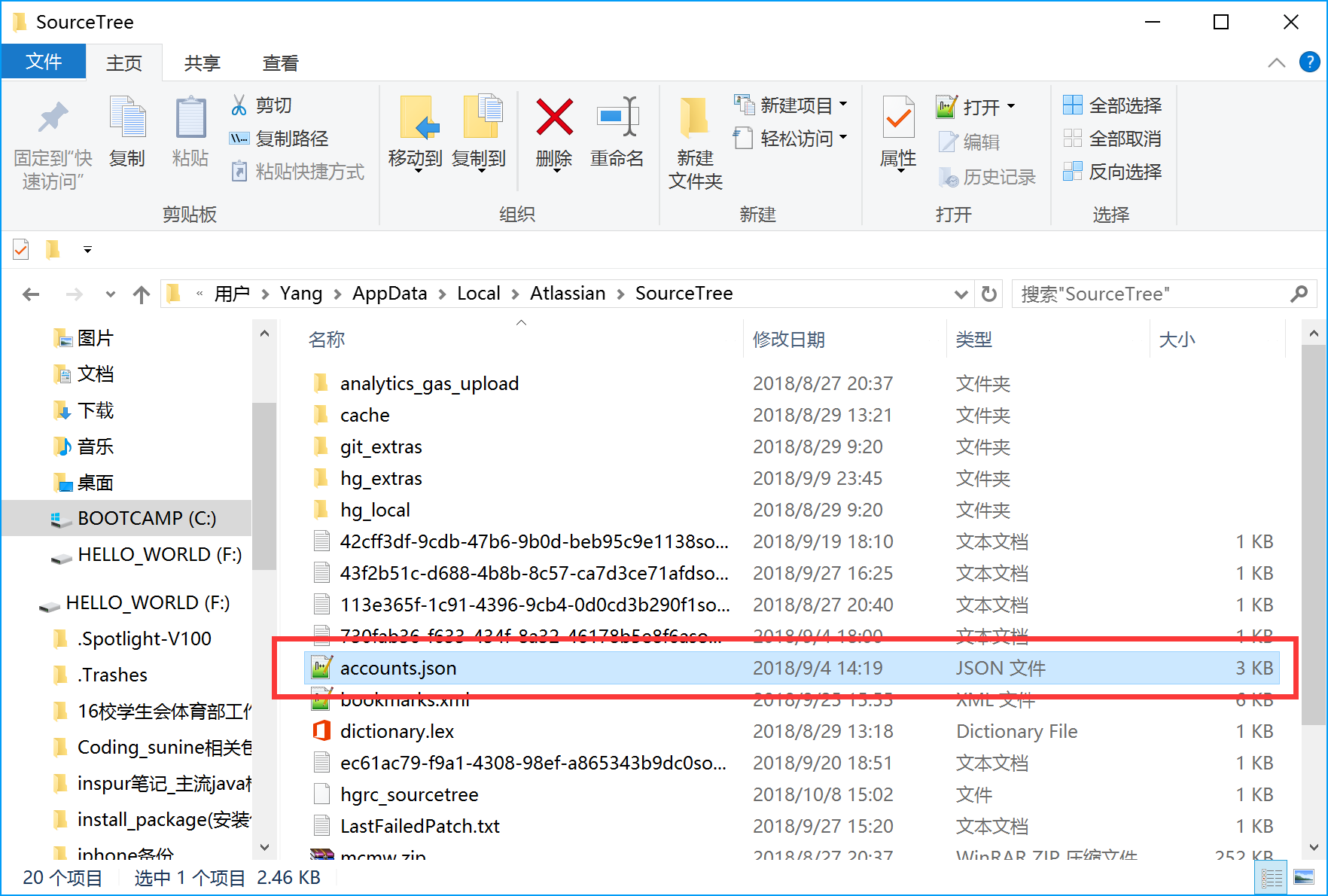


* + **全局配置：**由于主要使用Sourcetree图形化控制，所以Git全局配置不做赘述。可自行参考资料键入命令配置。

## Sourcetree的安装和配置

* + **须知：**Sourcetree下载地址服务器在境外，可能需要VPN或者较高网速访问。
  + **下载：**登录官网下载(<https://www.sourcetreeapp.com/>)。
  + **安装：**默认安装选项即可。安装完成会出现注册界面，可以连接到注册界面注册即可进入。如网络条件有限，可修改账户的json文件：
    - * + 地址栏键入：

%LocalAppData%\Atlassian\SourceTree\



* 打开accounts.json文件，输入以下内容，保存并重新打开sourcetree：

[

{

"$id": "1",

"$type": "SourceTree.Api.Host.Identity.Model.IdentityAccount, SourceTree.Api.Host.Identity",

"Authenticate": true,

"HostInstance": {

"$id": "2",

"$type": "SourceTree.Host.Atlassianaccount.AtlassianAccountInstance, SourceTree.Host.AtlassianAccount",

"Host": {

"$id": "3",

"$type": "SourceTree.Host.Atlassianaccount.AtlassianAccountHost, SourceTree.Host.AtlassianAccount",

"Id": "atlassian account"

},

"BaseUrl": "https://id.atlassian.com/"

},

"Credentials": {

"$id": "4",

"$type": "SourceTree.Model.BasicAuthCredentials, SourceTree.Api.Account",

"Username": "",

"Email": null

},

"IsDefault": false

}

]

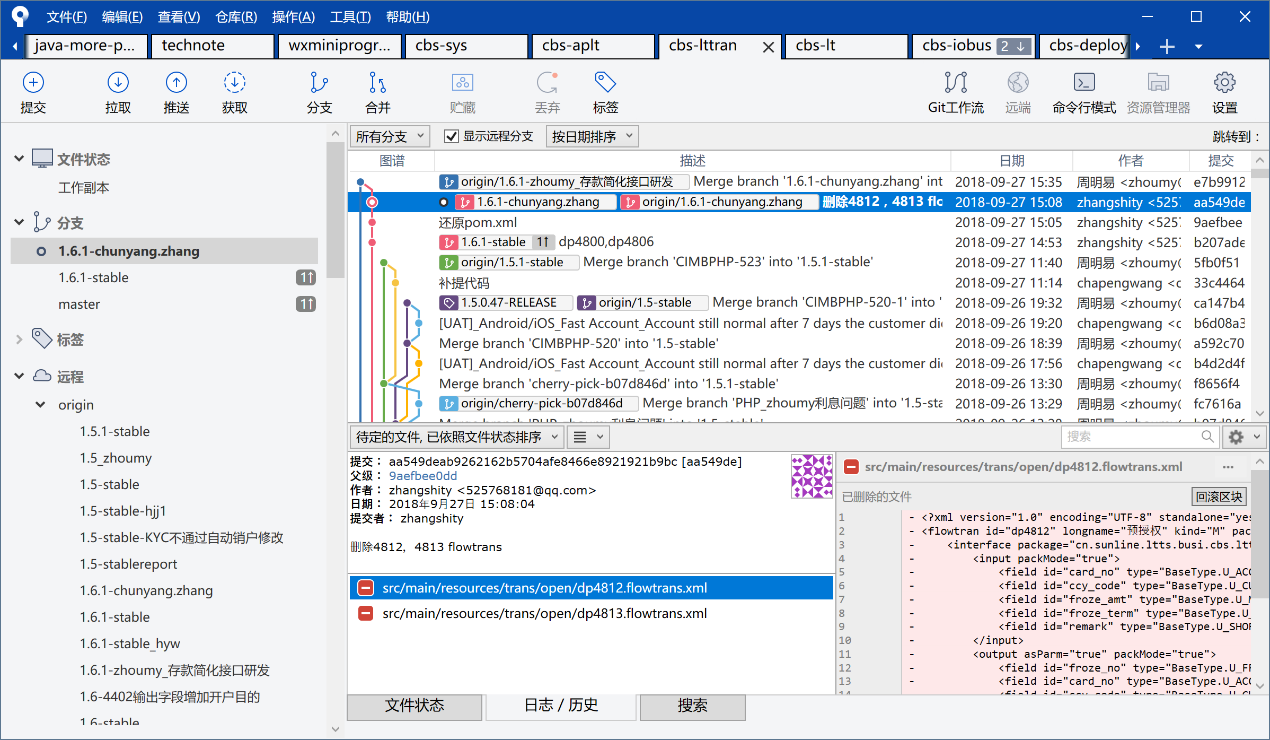
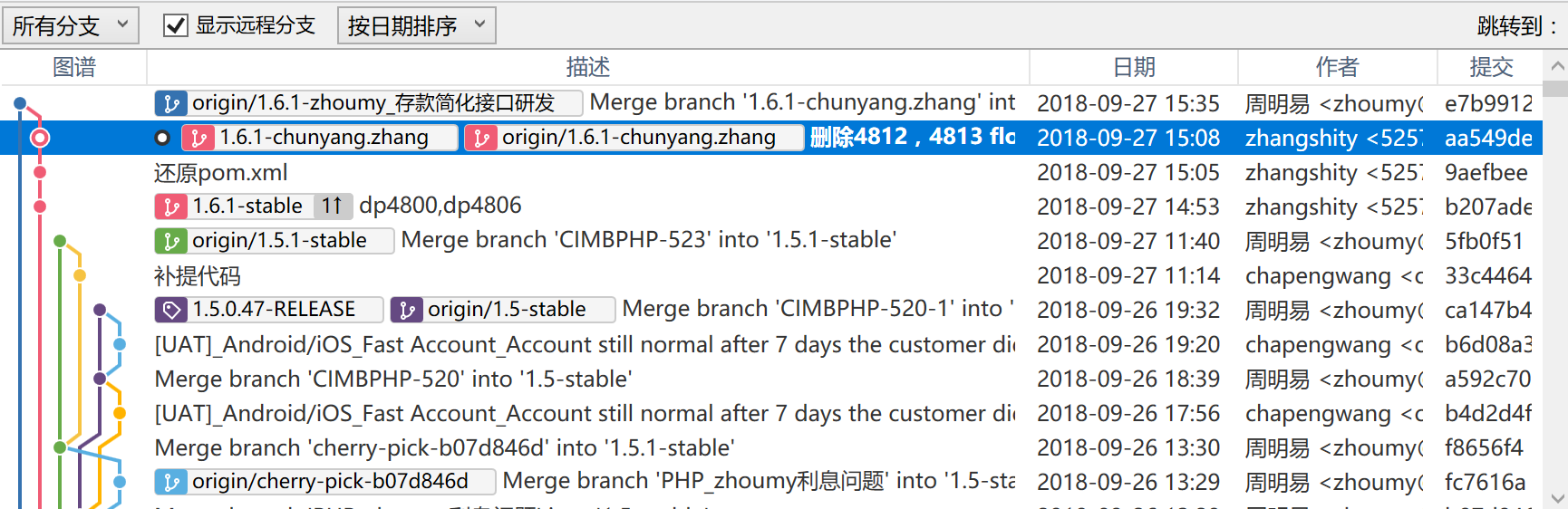
# 操作和使用

## 界面概览

* **须知**：需要掌握Git的基础知识和专业术语。

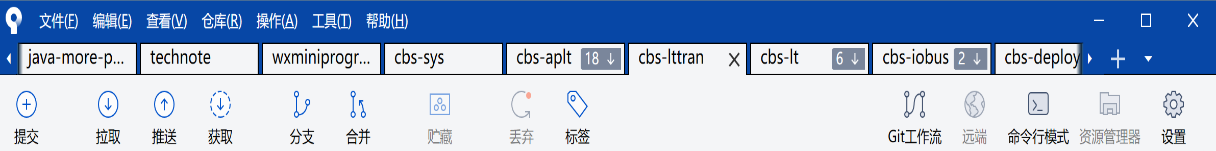
菜鸟：http://www.runoob.com/git/git-tutorial.html

w3cshool：https://www.w3cschool.cn/git/

* **概览**：主要的工作空间界面。
* 图谱工具：提交(commit)记录可视化，时间、作者、分支版本等一目了然，清晰明了。
* 分支可视化：本地(local)和远程(promote)分支同时可视。方便操作对比，如检出(check out)操作。



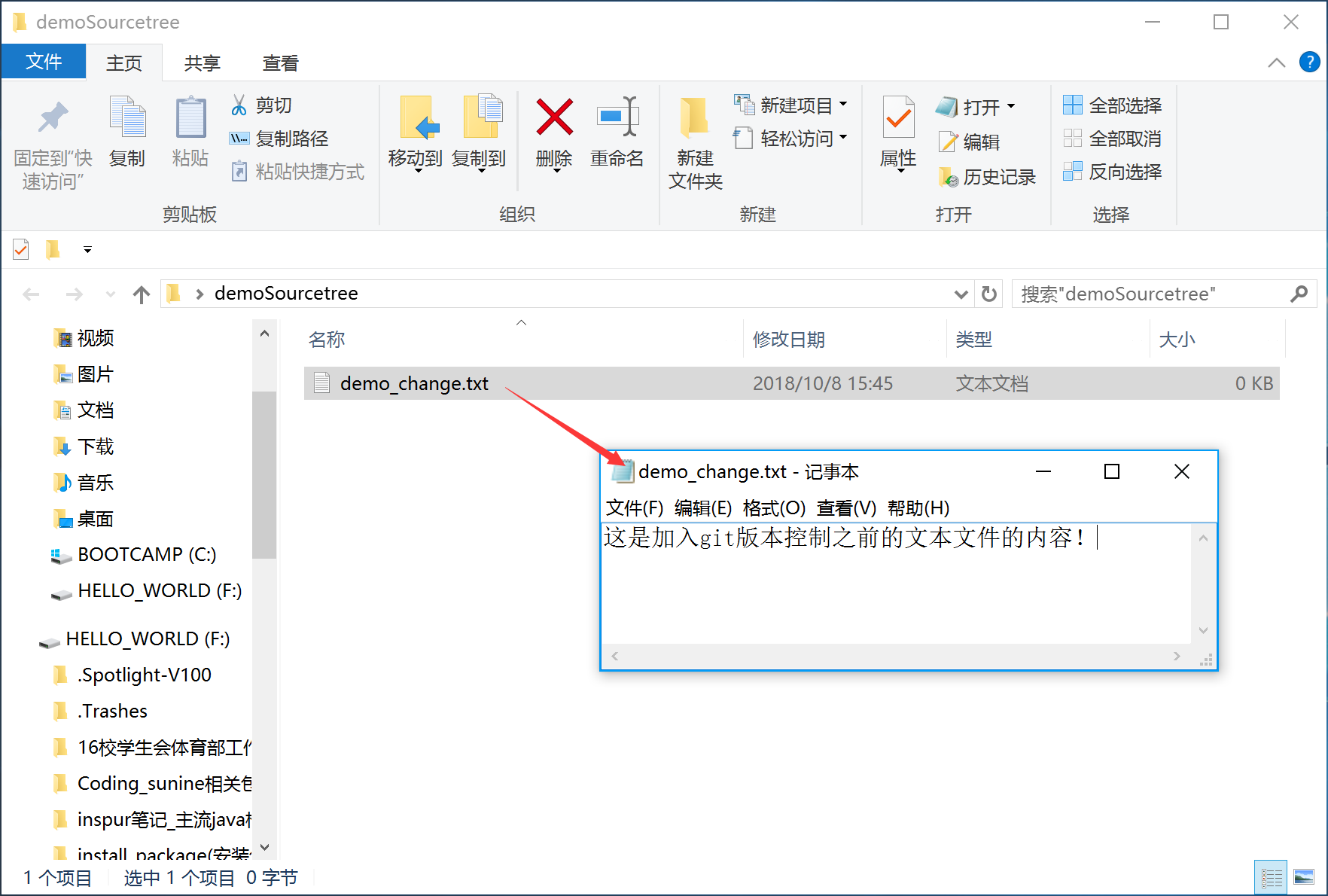
* 常用命令按钮：快捷操作。



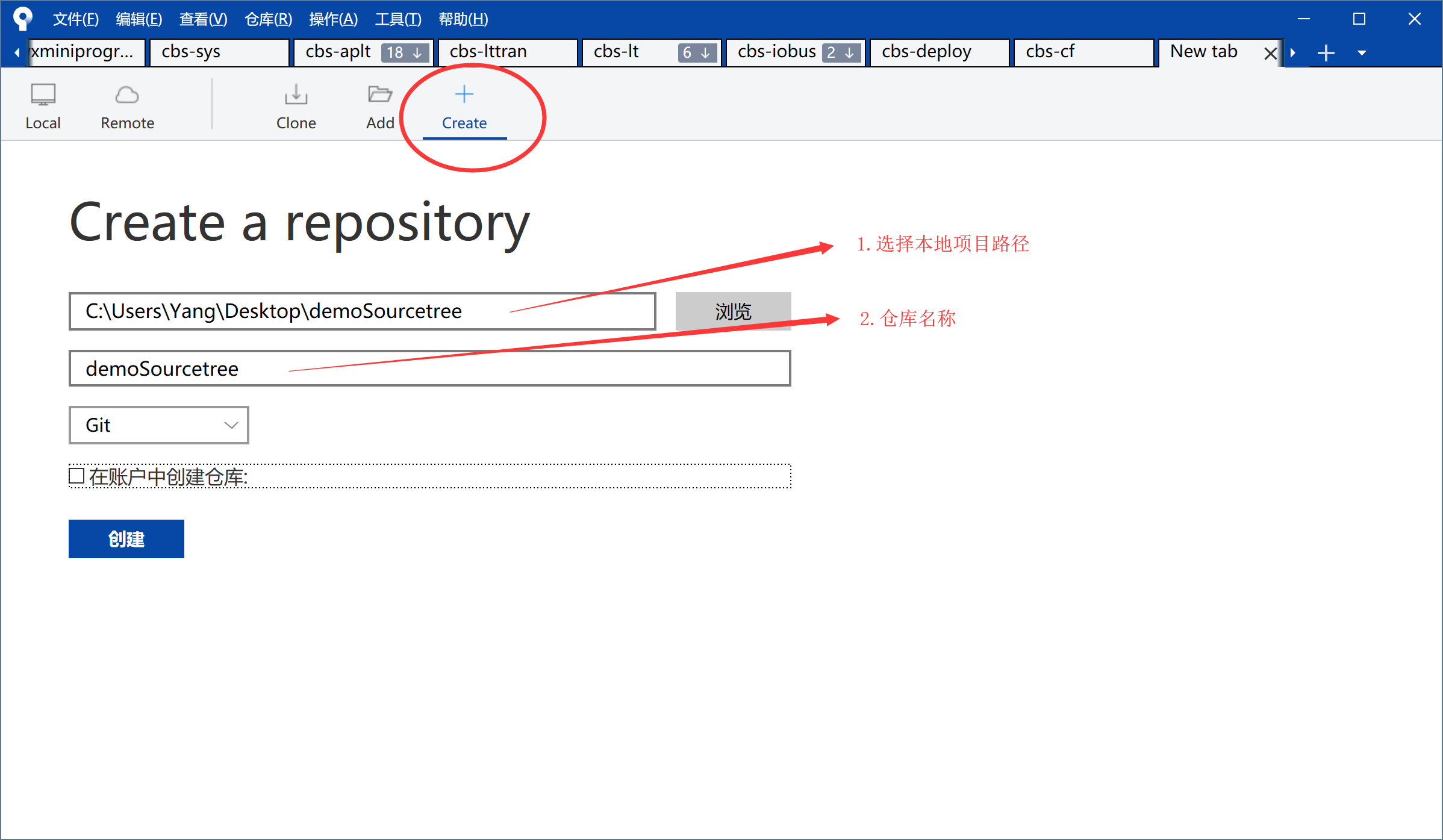
## 新建仓库

### 新建本地仓库

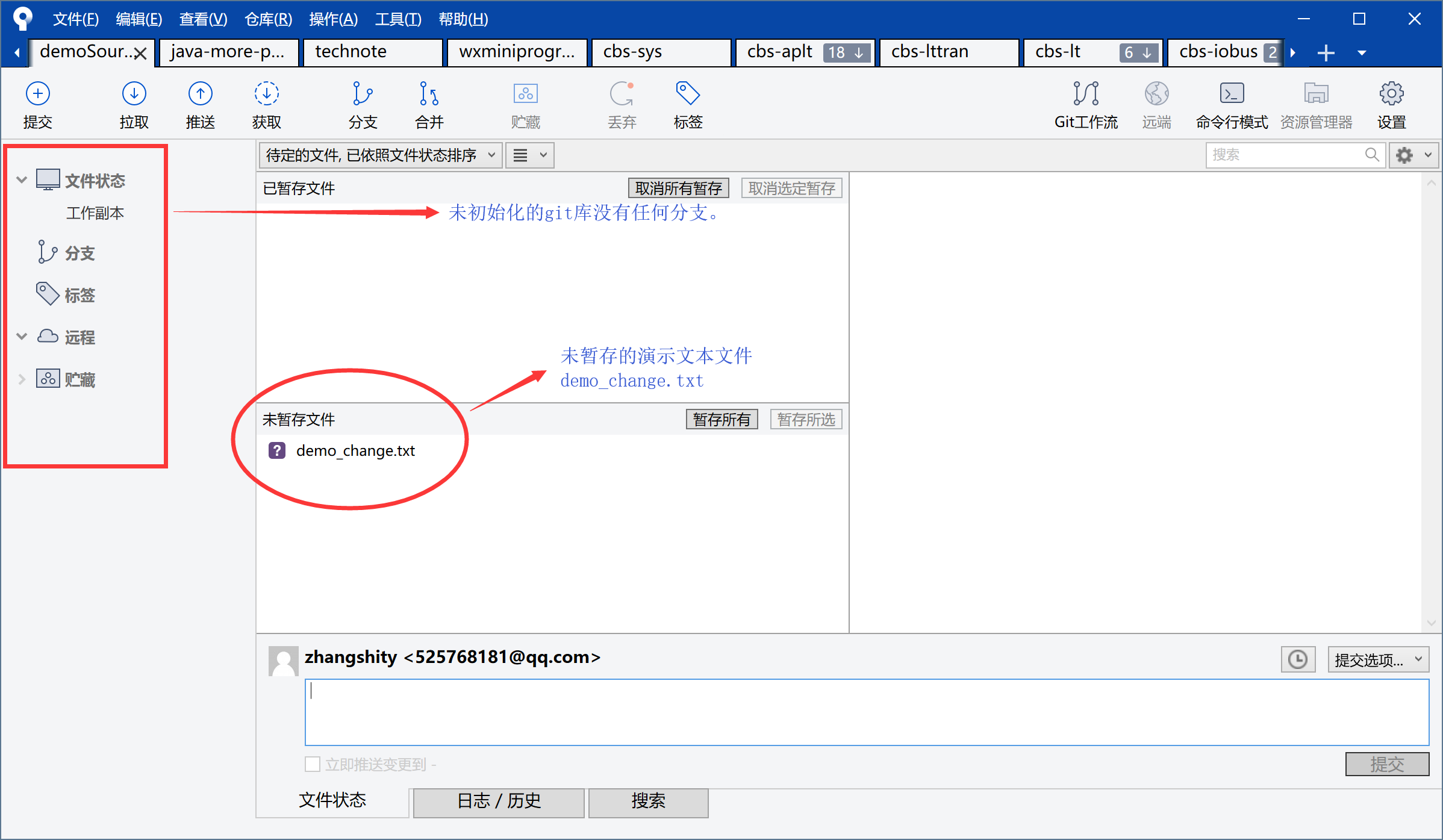
* 1.创建演示项目文件夹，新建一个txt文本文件，输入字符。



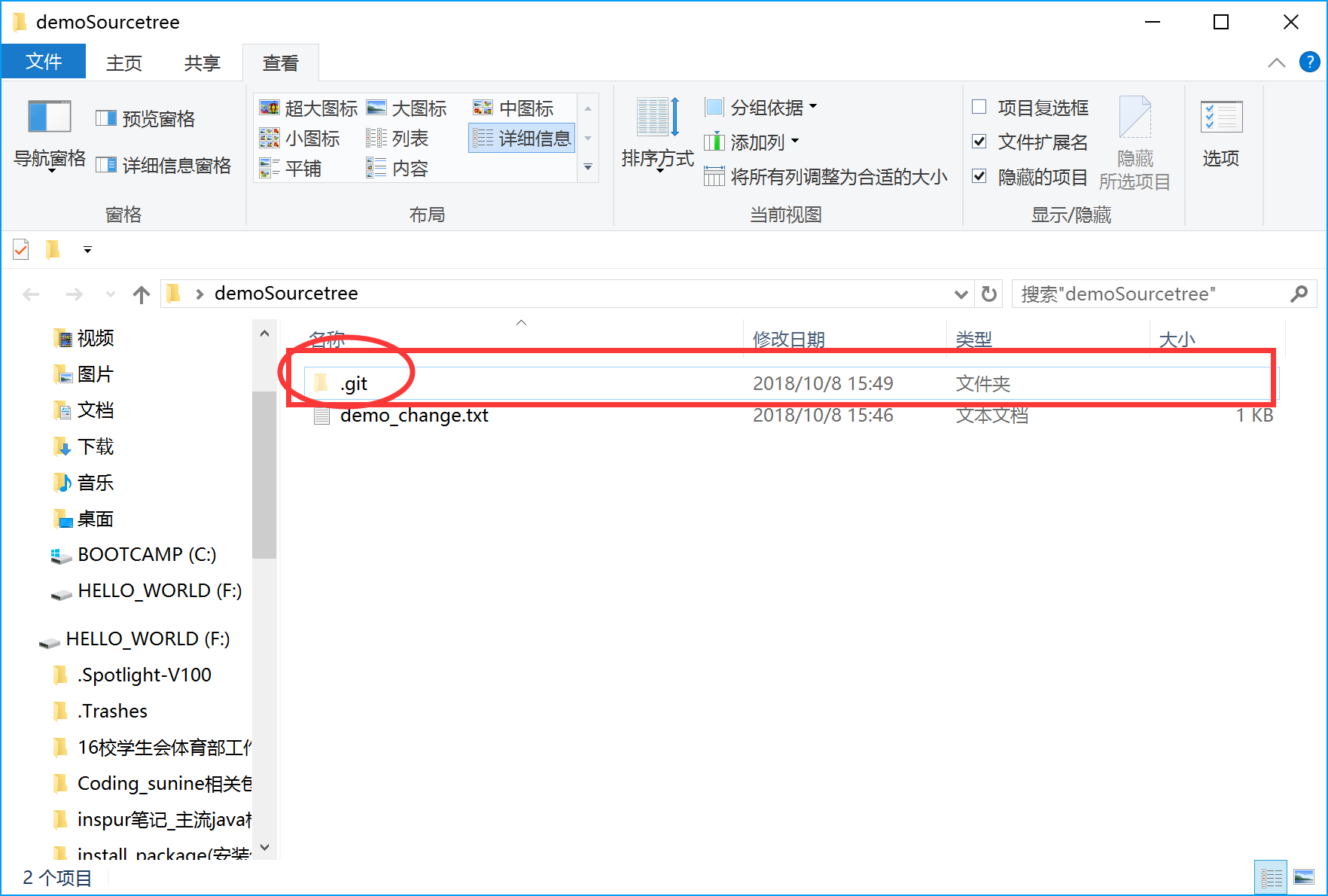
* 2.打开Sourcetree，点击create按钮创建新的本地仓库，选择项目文件并键入仓库名称。



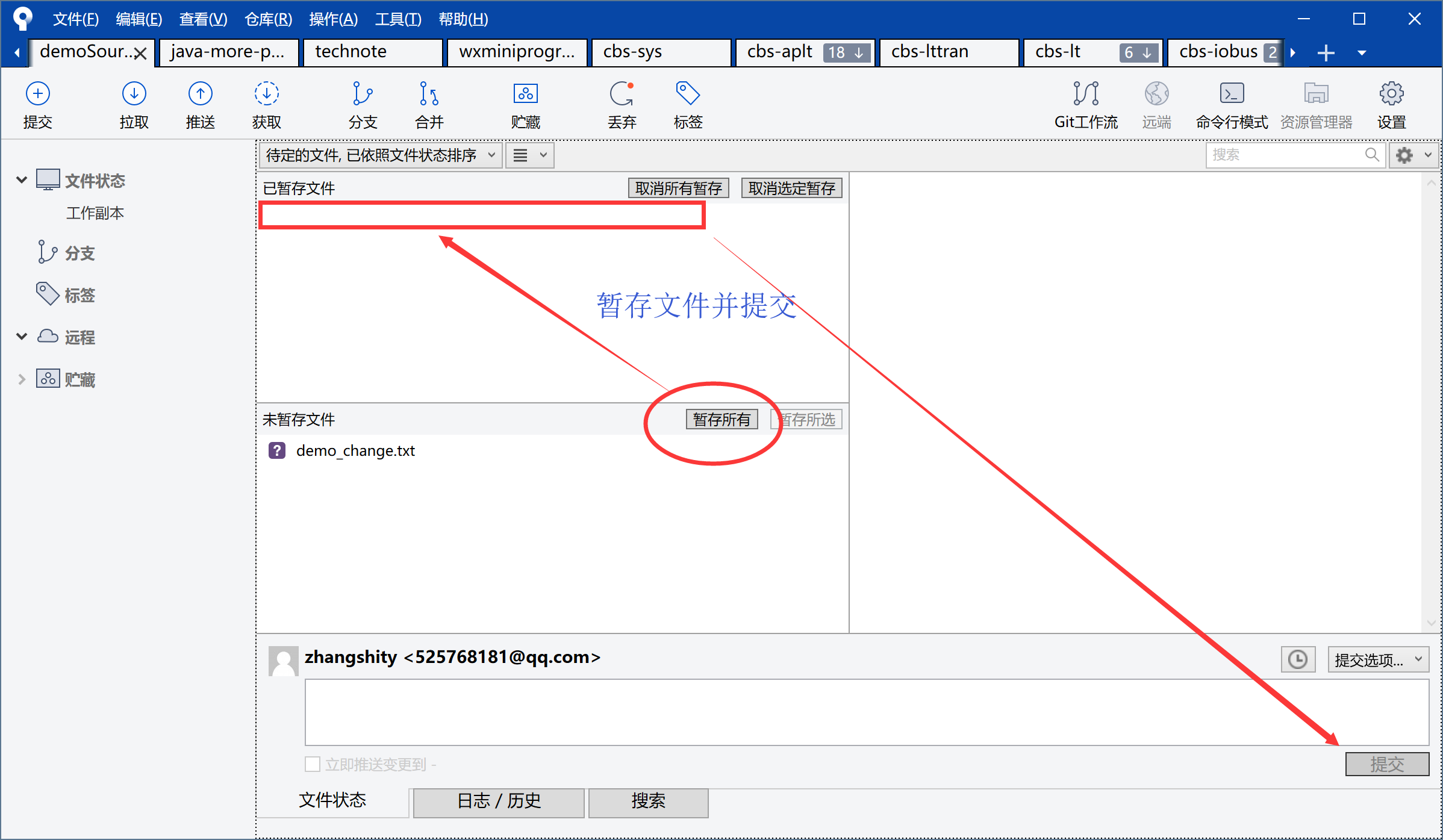
* 3.对比sourcetree和文件的不同
* Sourcetree界面里：新的仓库没有任何分支，没有任何内容，只有一个为暂存的演示文本文件。(因为没有初始化git仓库)



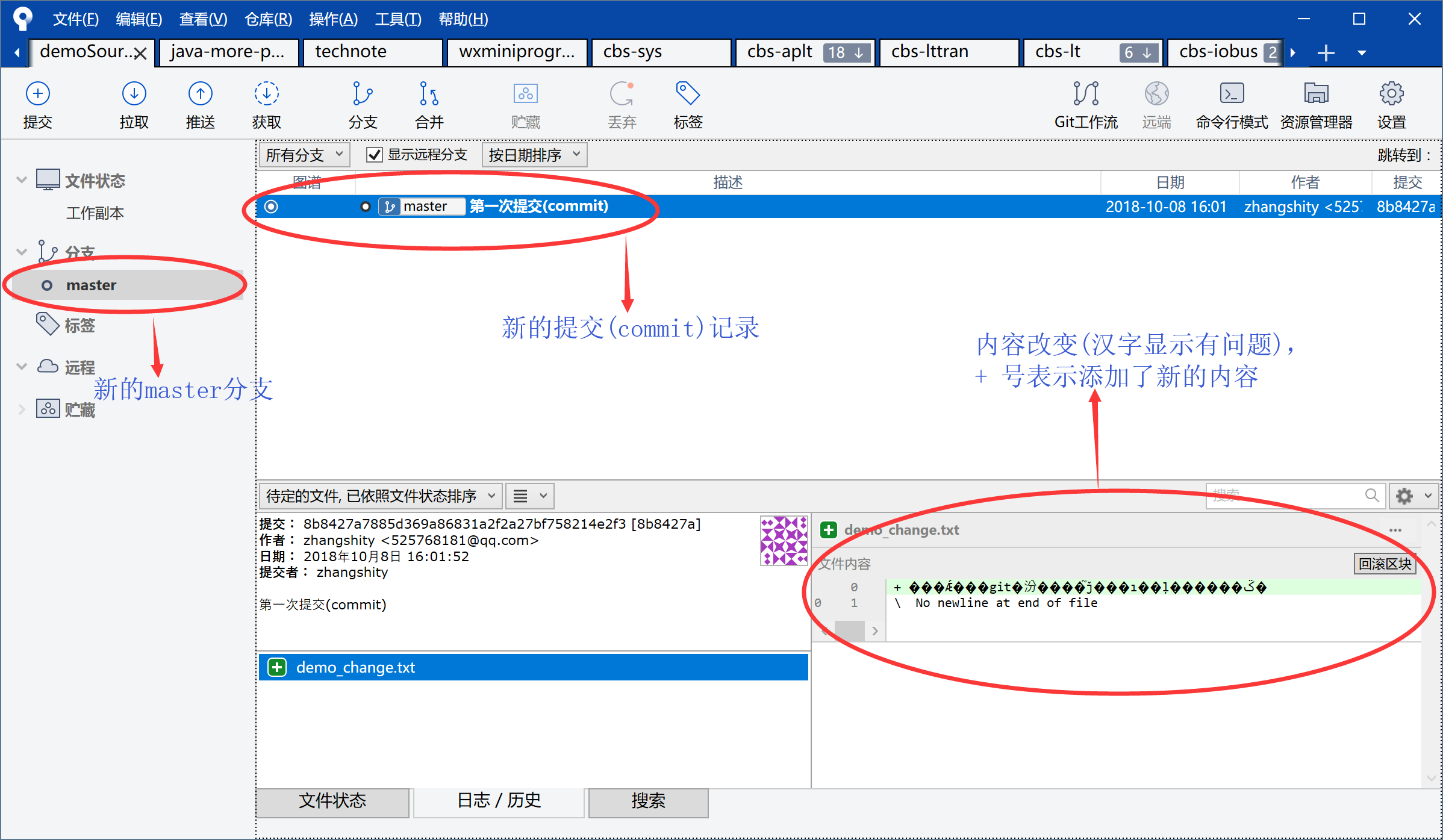
* 文件夹里：多了一个.git的隐藏文件。(代表git仓库创建成功)



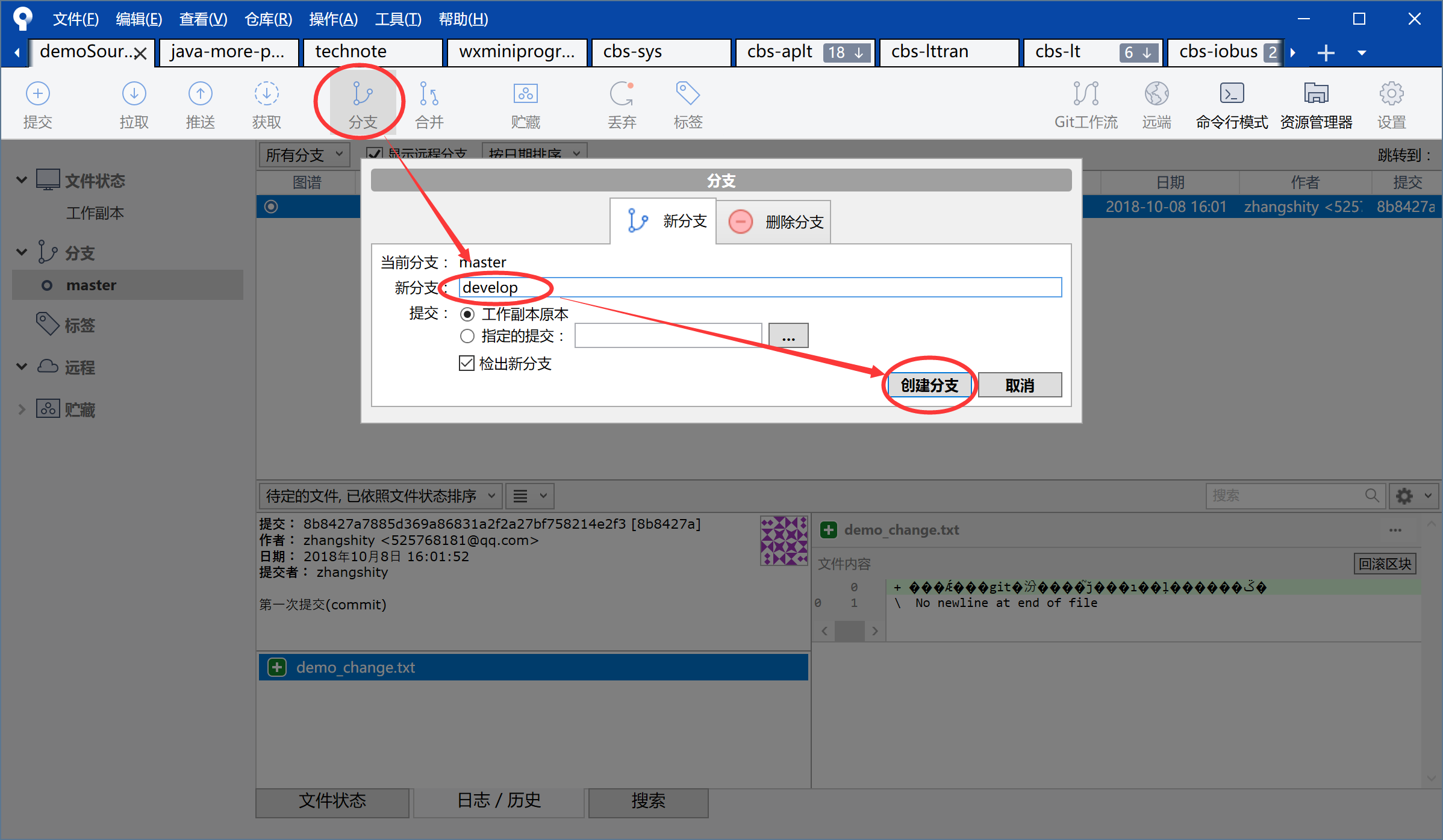
* 4.初始化git仓库，并把新的改动demo\_change.txt加入git版本控制中。

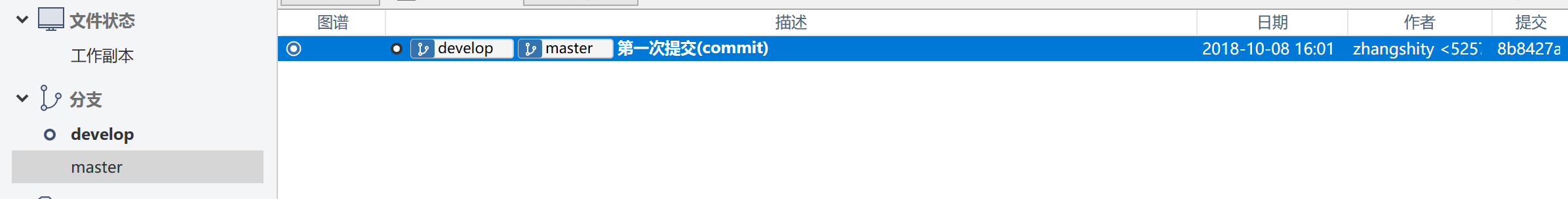


* 5.完成git仓库的初始化，对比提交后的变化。



* 6.创建新的开发分支(develop)，sourcetree会在master分支的基础上创建一个内容完全一样的分支，并自动切换到此develop分支，方便开发者操作。



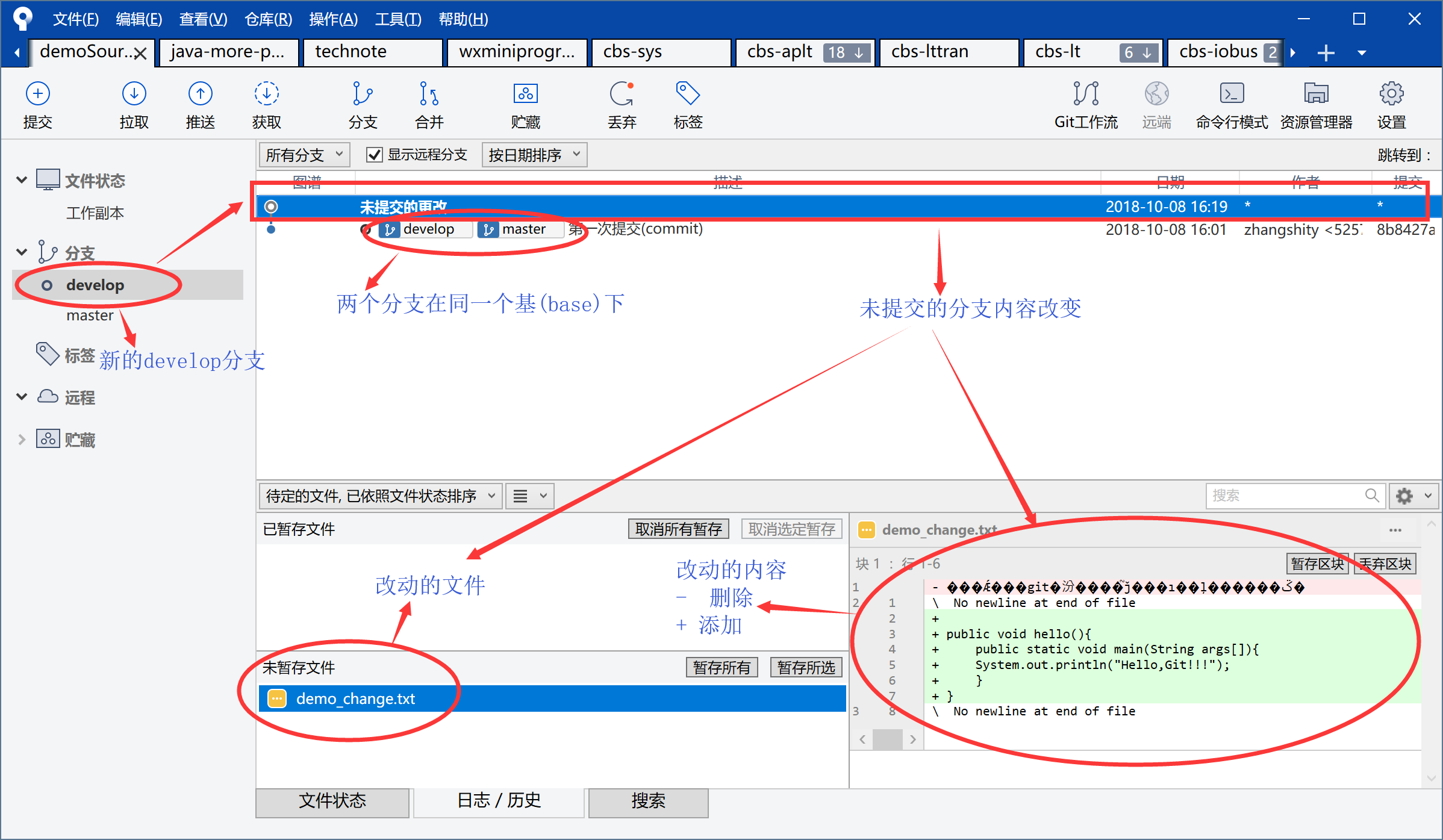


* 7.在develop分支上做改动：打开之前编辑的文本，删除内容，添加新内容，并保存。

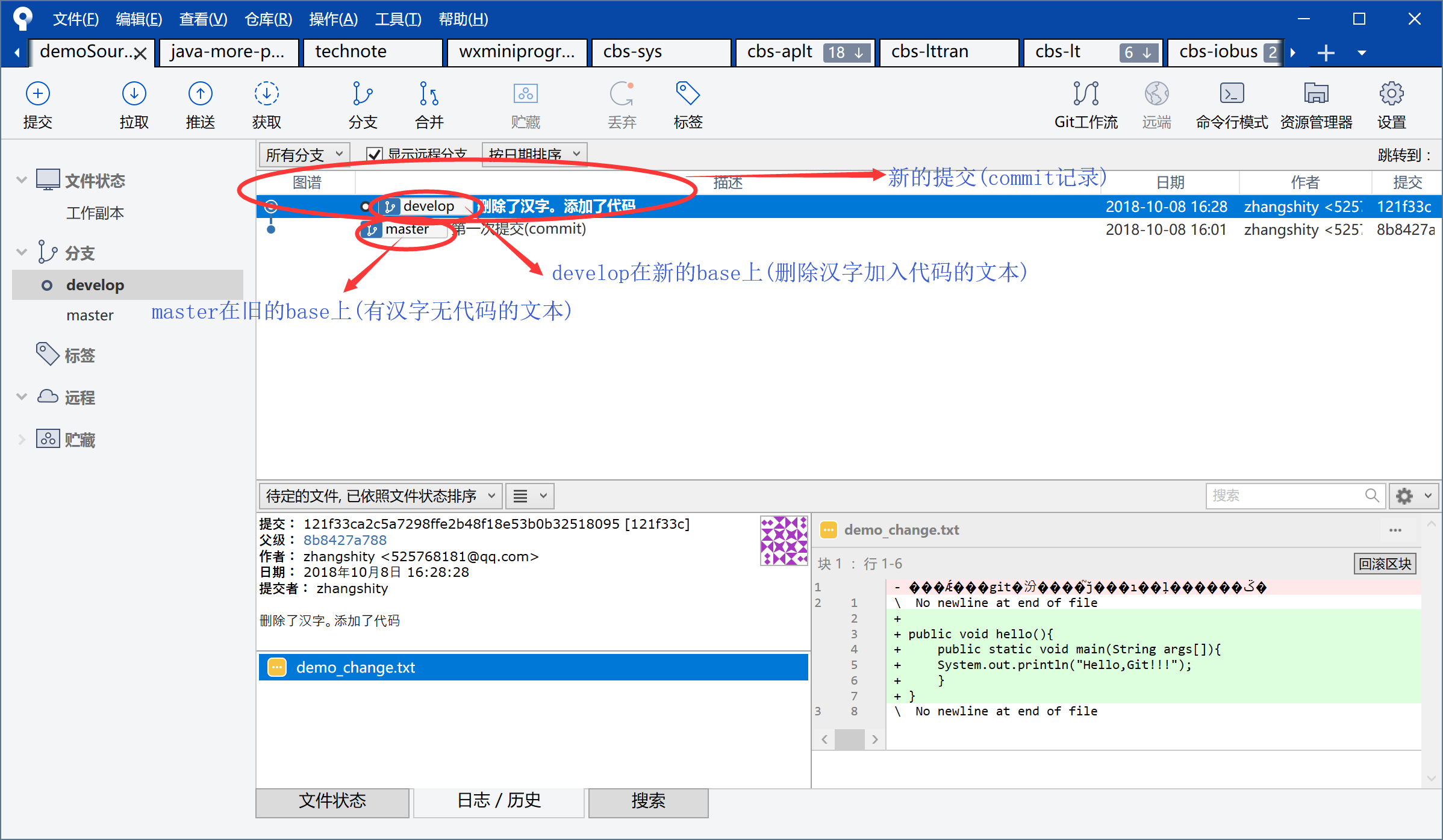
完成后切换回sourcetree，对比改变，然后在当前的develop分支下提交新的改变(提交过程与步骤相同)。

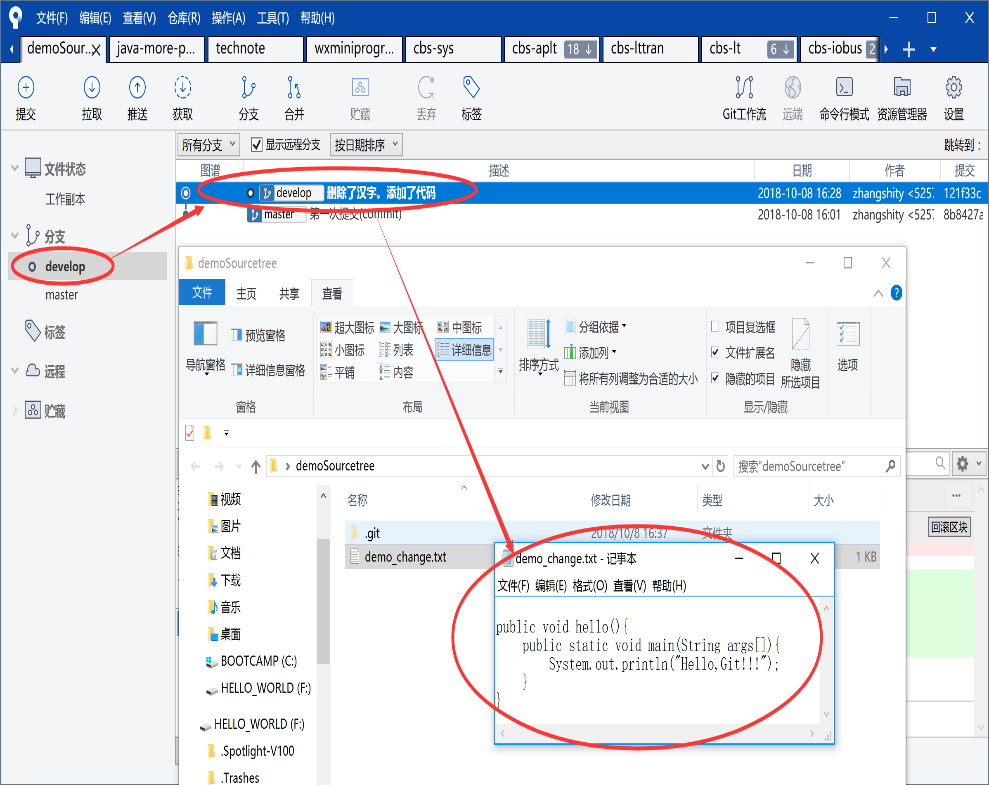
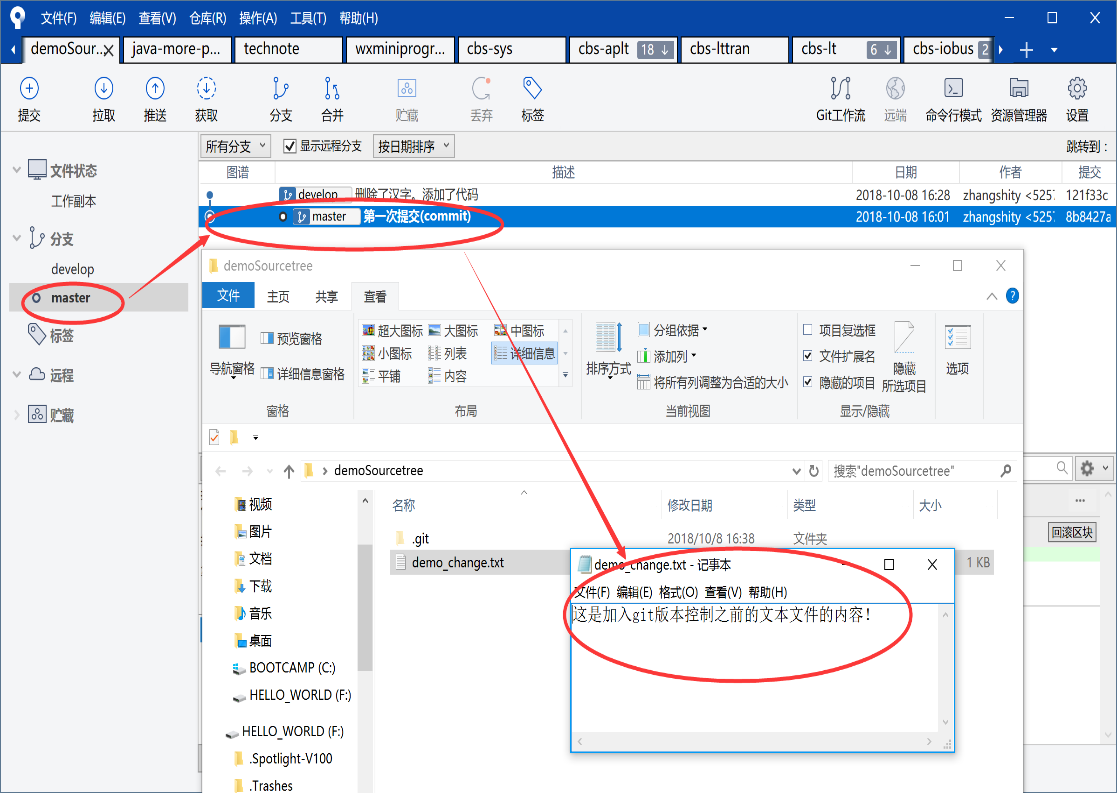


更改完文本。提交(commit)更改



提交完后对比提交变化。



双击切换分支，发现文本内容，在不同的分支中不同。

* 8.同时可以创建远程仓库(github、gitlab、bitbucket)，并把代码推送(push)到远程，或从远程拉取(pull)下来。

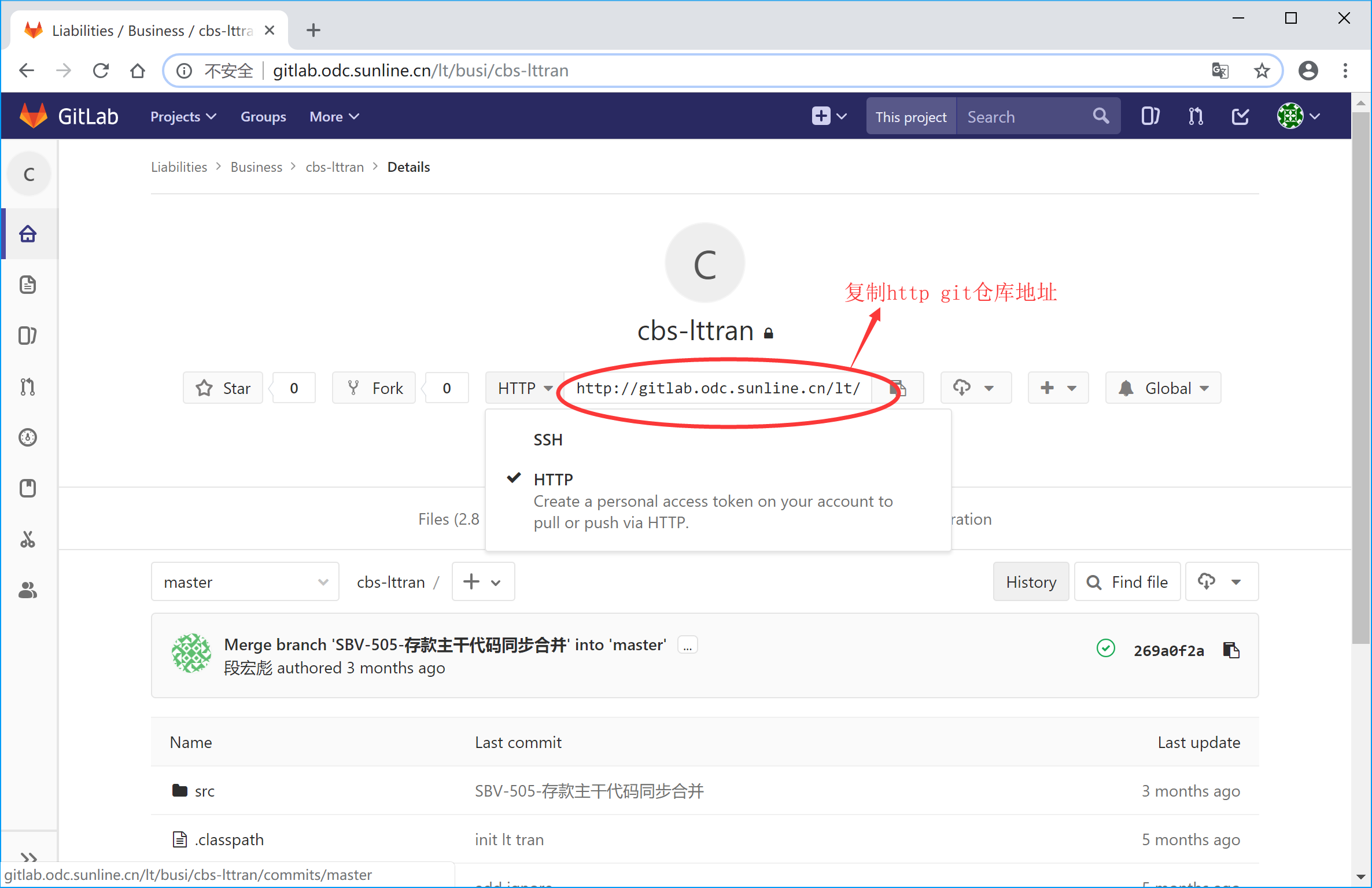
略…….

### 克隆远程仓库

* 1.远程仓库拷贝http或ssh的git仓库地址

(企业内部为了安全考虑一般使用SSH协议。)

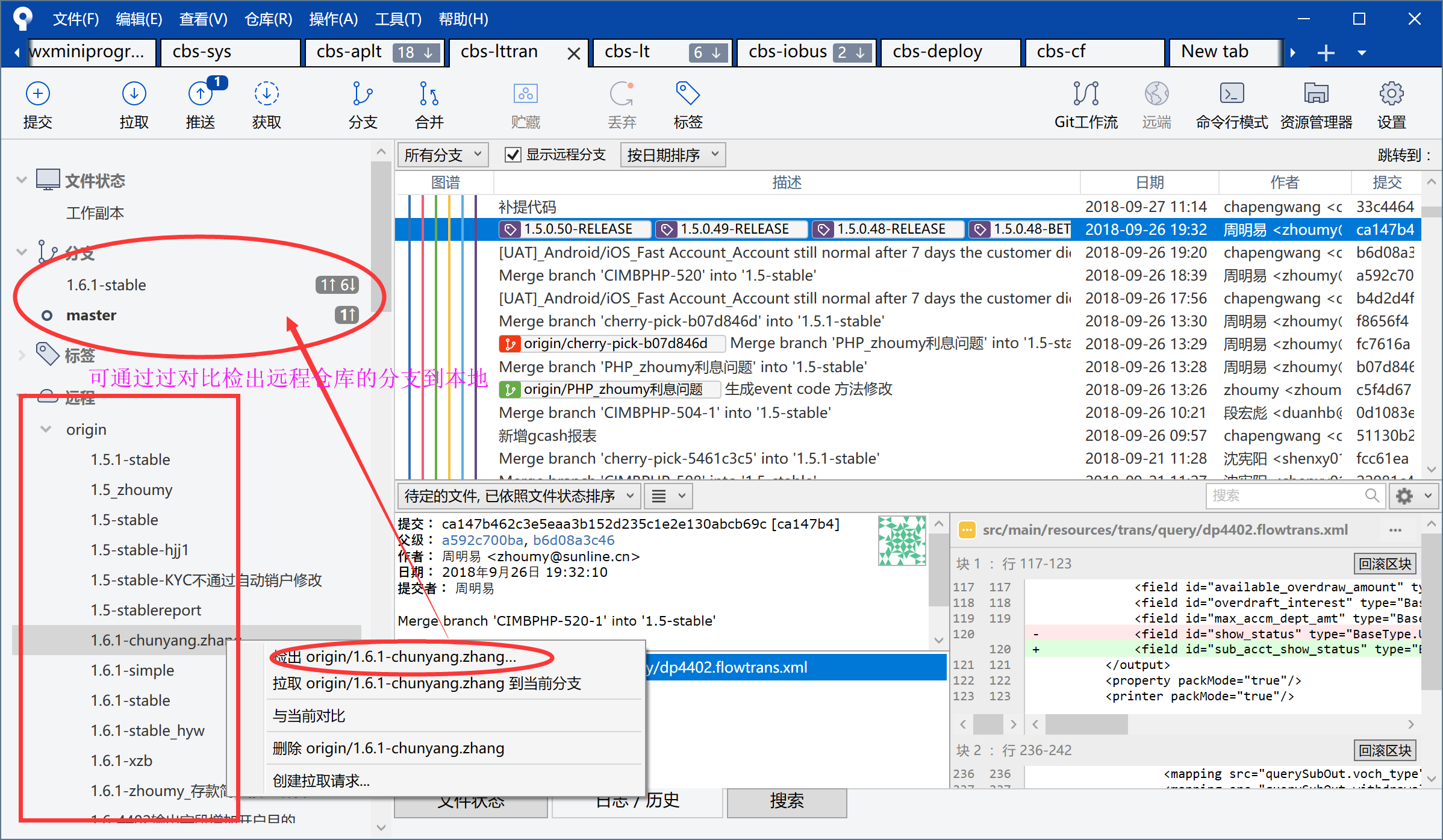
(这里只介绍简单的http协议的克隆。SSH协议相同，只是多了一个本地生成ssh-key的步骤和远程添加ssh-key的步骤，有需要会补充修改。)



* 2.添加远程仓库地址到本地客户端



* 3.克隆完成，检出远程分支到本地，以便对代码进行修改。





* 4.在IDE修改代码，完成后推送到远程同样的分支。在远程仓库提交合并请求，等待合并。

