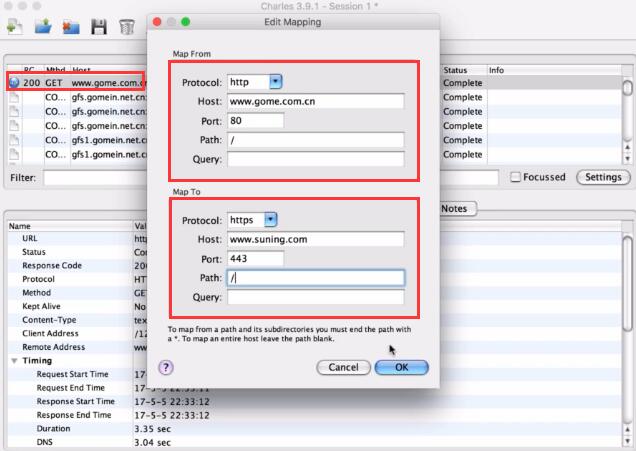


1. 在命令行工具中输入node -v 检查安装node.js的版本及是否安装

安装代理服务器 Charles/Fiddler

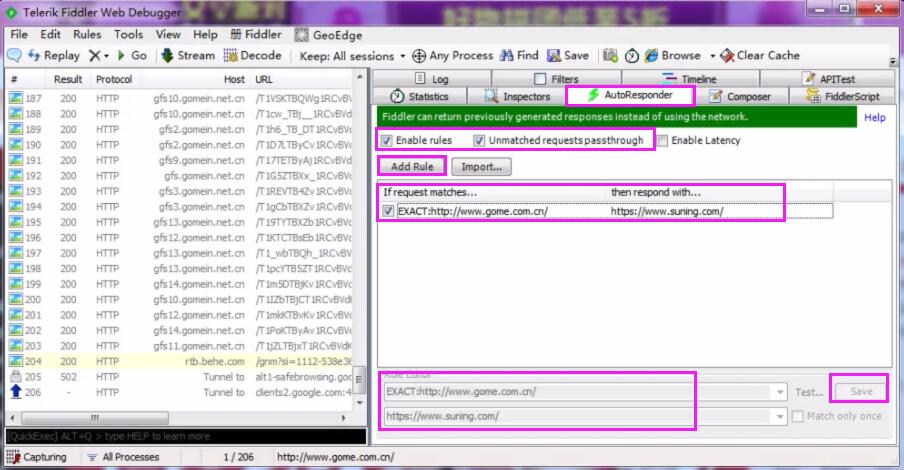
1.Charles的用法 案例：

找到要被代理的网址右键——Map Remote填写信息 Protocol:http Host:www.gome.com.cn Port:80 Path:/ Mapto的填写Protocol:https Host:www.suning.com.cn Port:443 Path:/ ——点击OK



2.Fiddler的用法。案例：

找到要代理的网址——点击右侧的AutoResponder,选中Enable rules 和Unmatched requests passthrough 点击Add Rule，然后在Rule Editor 处填入相应网址和信息 点击右侧save按钮。



Git的安装和.gitconfig配置

1.网址：<https://git-scm.com/>

2.开始——程序——Git-Bash，输入dir回车，输入ls回车，输入git --version回车，在git Bash根目录下输入 vim .gitconfig点击回车(这一系列步骤快捷键：vim ~/.gitconfig)

3.点击键盘的i键，复制git的配置，shift+ins粘贴。按Esc ，输入:wq按回车保存退出。

（这个目的是让git的命令简介化，也可以不进行配置）

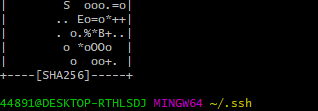
SSH密钥的配置

1. 开始——程序——Git Bash——cd ~/.ssh检查下这台电脑有没有配置SSH 如果出现这个下图所示，表明该电脑还没配置SSH

则需要进行下面的操作



1. 粘贴 ssh-keygen -t rsa -C "448919239@qq.com" 多次回车（4次）直到命令行出现。如图所示;



注意:这里的操作不是一层不变的。

3.44891@DESKTOP-RTHLSDJ MINGW64 ~/.ssh目录下输入ls可见到id\_rsa id\_rsa.pub这两个文件夹，在44891@DESKTOP-RTHLSDJ MINGW64 ~/.ssh后输入cat id\_rsa.pub后回车既可以看到公钥。

4.复制

ssh-rsaAAAAB3NzaC1yc\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*448919239@qq.com

gitLab仓库的建立

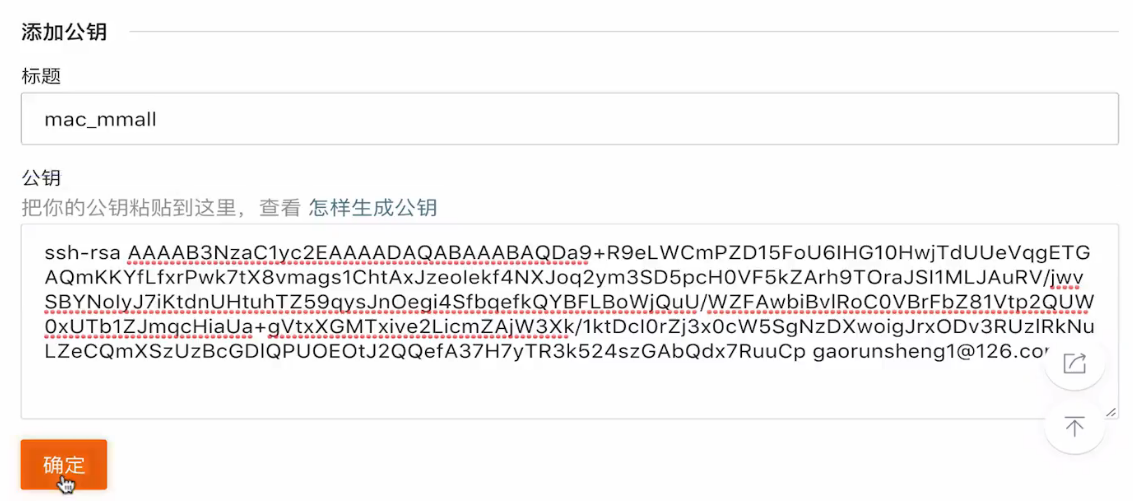
1.回到git托管中心gitee.com，网址：<https://gitee.com/>

2.新建一个yqfhappymmall组织。

3.添加一个项目，名称：mmall-fe，选择语言：JavaScript, 添加.gitignore 添加开源许可证 是否公开：私有 ——创建



1. 找到“管理”——“部署公钥管理”——“公钥管理”——点击“添加个人公钥”——填好公钥名称（例如：MAC）



6.把复制的

ssh-rsaAAAAB3NzaC1yc\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*448919239@qq.com 粘贴到公钥输入框——点击“确定”。则SSH密钥的配置就完成了

从远程仓库代码拉取到本地电脑

1.自己手动在自己的电脑喜欢的位置建个doc文件夹（也可以用快捷键mkdir doc创建）

2.通过Git Bash进入doc文件夹,

输入git clone git@gitee.com:yqfhappymmall/mmall-fe.git回车出现这个代表成功

Cloning into 'mmall-fe'...

remote: Counting objects: 3, done.

remote: Total 3 (delta 0), reused 0 (delta 0)

Receiving objects: 100% (3/3), done.

3.输入ls回车可以查看git仓库中的mmall-fe文件就被拉取到本地了，输入cd mmall-fe进入到mmall-fe文件夹，接下来就可以做代码开发了。(这时候就会自动多一个README.ms的文件)

将本地的文件夹push到git远程

1.第一次上传一定要初始化项目，输入git init 回车 进行初始化 目的是在本地创建一个暂存区。

2.输入git remote add origin git@gitee.com:yqfhappymmall/mmall-fe.git 回车，使本地暂存区的内容和远程仓库建立关系

（第一次上传代码时前两步必须有，之后再上传可以直接从第三步开始）

3.git pull origin master 有联系了还不行需要把远程的分支拉一下master分支上的代码到本地 回车 出现如下这个

remote: Counting objects: 3, done.

remote: Total 3 (delta 0), reused 0 (delta 0)Unpcking objects: 100% (3/3), done.From gitee.com:yqfhappymmall/mmall-fe

\* branch master -> FETCH\_HEAD

\* [new branch] master -> origin/master

4.输入git status 查看变化的文件

5.输入git add .把当前的文件追踪上来

6.输入git commit -am '注释' 将全部文件提交到暂存区，并加上注释

7.输入git push 把本地的东西推送到远程服务器上 首次提交会出现了以下这个

fatal: The current branch master has no upstream branch.

To push the current branch and set the remote as upstream, use

git push --set-upstream origin master

8.输入git push --set-upstream origin master 回车 这样本地的代码就上传到远程的仓库

.gitignore的配置

1.MINGW64 ~/mmall/doc/mmall-fe (master)\*$后输入vim .gitignore 在项目根目录生成一个.gitignore文件 进行.gitignore的配置 回车

2.按一下i键，输入.DS\_Store /node\_modules/ /dist/ \*.log npm-debug.log 等之后进行Esc ：wq回车保存。

切换分支开发

1. git checkout -b mmall\_v1.0

通过git建立文件夹

1. 在分支~/mmall/doc/mmall-fe (mmall\_v1.0) $ ls查看确保该分支没有杂七杂八的文件。如果有就输入rm -rf 和文件名给其删掉（例如：rm -rf dist/）
2. 输入git st检查新分支有没有其他没用的文件 出现以下这块就可以正常建目录了 （建五个最基础的文件夹）

On branch mmall\_v1.0

nothing to commit, working tree clean

1. MINGW64 ~/mmall/doc/mmall-fe (mmall\_v1.0) $ mkdir src
2. MINGW64 ~/mmall/doc/mmall-fe (mmall\_v1.0) $ cd src/
3. MINGW64 ~/mmall/doc/mmall-fe/src (mmall\_v1.0) $ mkdir view
4. MINGW64 ~/mmall/doc/mmall-fe/src (mmall\_v1.0) $ mkdir page
5. MINGW64 ~/mmall/doc/mmall-fe/src (mmall\_v1.0) $ mkdir service
6. MINGW64 ~/mmall/doc/mmall-fe/src (mmall\_v1.0) $ mkdir util
7. MINGW64 ~/mmall/doc/mmall-fe/src (mmall\_v1.0) $ mkdir image

在依次输入git st,ls见到src文件夹

到这里项目目录的搭建工作就完成了，接下来就可以进行项目脚手架的搭建了

知识点：

git init 初始化项目，目的是将本地目录放入一个仓库(也可叫：本地的暂存区)

npm init初始化目的是输入参数，生成package.json（项目模块化的描述文件）

git add . 就是将本地修改过的文件且已经追踪的文件添加到本地的暂存区

git commit -am "注释" 将暂存区的代码提交到本地仓库

git checkout -b "分支名" 用来创建新分支并切换到该分支

git checkout "分支名" 切换到某一个已经存在的分支

git branch 查看当前的分支

git merge 分支的合并

git branch -a 查看所有的分支

git pull/git push 代码的拉取和提交

ls 查看文件夹，

rm test.html 将test.html删除

ls 是查看进入对应的文件夹所包含的子文件夹

cd .. 回到上一层

Ls -al 查看所有的文件

git b 查看分支

rm -rf dist/ 删除dist文件夹

mkdir src 新建一个src目录

cd src/ 进入src目录

touch .DS\_Store（建立个文件）（注意：创建文件夹是用mkdir，创建文件用touch）

输入git checkout -b mmall\_v1.0进行切分支开发 （加了个-b表示mmall\_v1.0还没有要创建） 输入git b查看分支

常用的git配置

[user]

name = yqf

email = 448919239@qq.com

[alias]

co = checkout

ci = commit

st = status

pl = pull

ps = push

dt = difftool

ca = commit -am

b = branch

[push]

default = simple

git commit -m 与 git commit -am 的区别

<https://blog.csdn.net/yaomingyang/article/details/78788053>