git&数据可视化-day02

黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

学习目标

Learning Objectives

1. 学习 jwt 知识，实现页面访问权限控制
2. 能够掌握 axios 请求和响应拦截器的配置
3. 能够掌握 Git 远程仓库操作的命令
4. 综合练习：实现项目推送远程仓库

目录

Contents

思考

在前后端分离模式的开发中，服务器如何知道来访者的身份呢？

基于token令牌（身份象征）

将军令在此，还不退下！

jwt身份认证 （json web token）

在前后端分离模式的开发中，服务器如何知道来访者的身份呢？

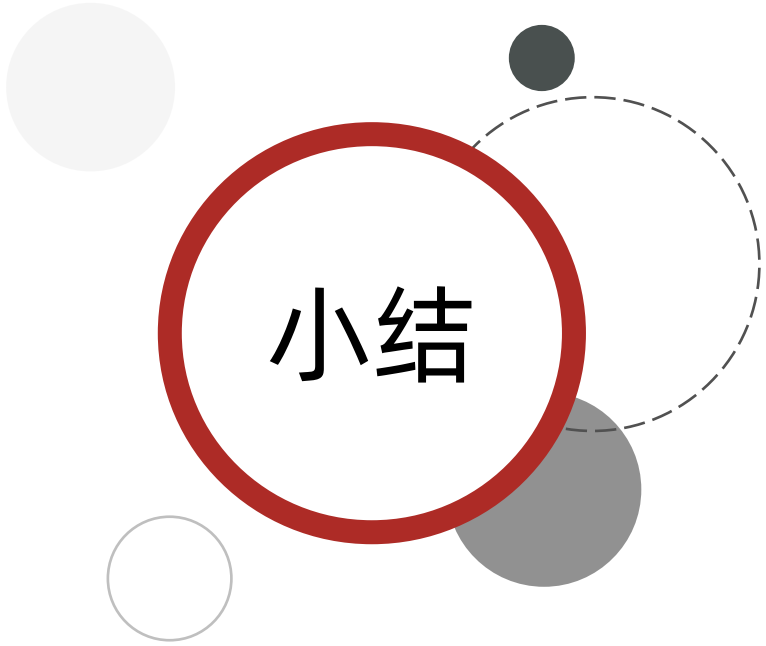
基于 token 令牌: 包含身份信息的字符串

jwt身份认证（json web token）

后端返回token，前端要存储到本地，下次请求时携带在请求头中！

```
document.querySelector('#btn-login').addEventListener('click', async function() {
  const data = serialize(document.querySelector('form'), { hash: true })
  // 非空校验
  if (!data.username) return tip('请输入用户名')
  if (!data.password) return tip('请输入密码')
  if (data.password.length < 6) return tip('密码的长度不能小于6位')

  try {
    const res = await axios.post('/login', data)
    localStorage.setItem('user-token', res.data.data.token)
    localStorage.setItem('user-name', res.data.data.username)
    tip('登录成功')
    location.href = './index.html'
  }
  catch (e) {
    tip('用户名或者密码错误')
  }
})
```



什么是 jwt ?

是一个基于 token令牌 的身份认证机制

登录请求后，后台返回token，前端该如何操作？

将token存储于本地，下次请求时在请求头携带

思考

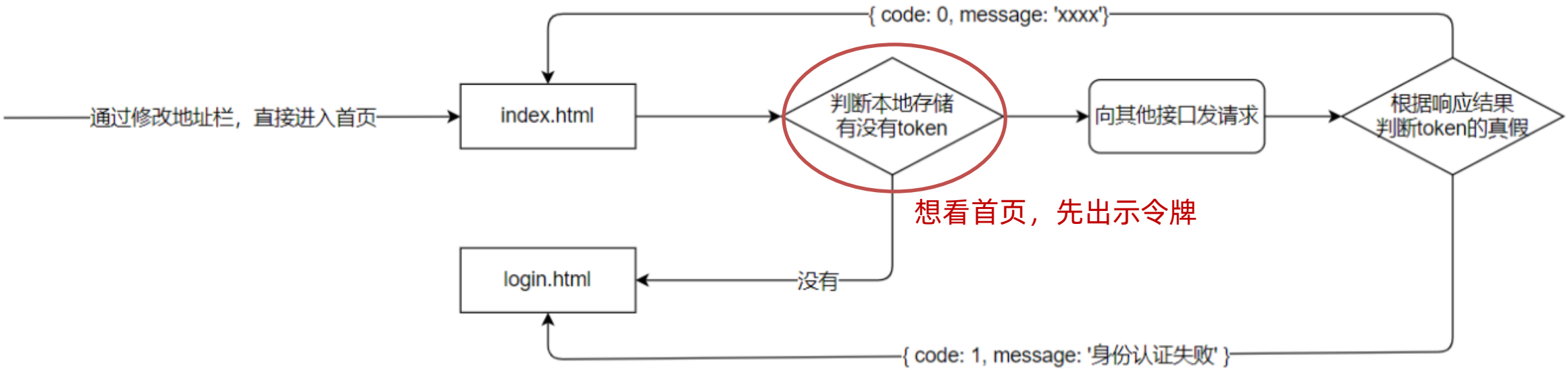
未登录的用户可以直接访问首页么？

如何判断有没有登录？ 判断是否有token令牌

皇宫重地，岂能擅入！请出示令牌！

页面访问拦截

浏览器端，可以通过合理使用令牌，控制页面的访问权限。



页面访问拦截：判断本地存储是否有token, 没有token拦截到登录

页面访问拦截

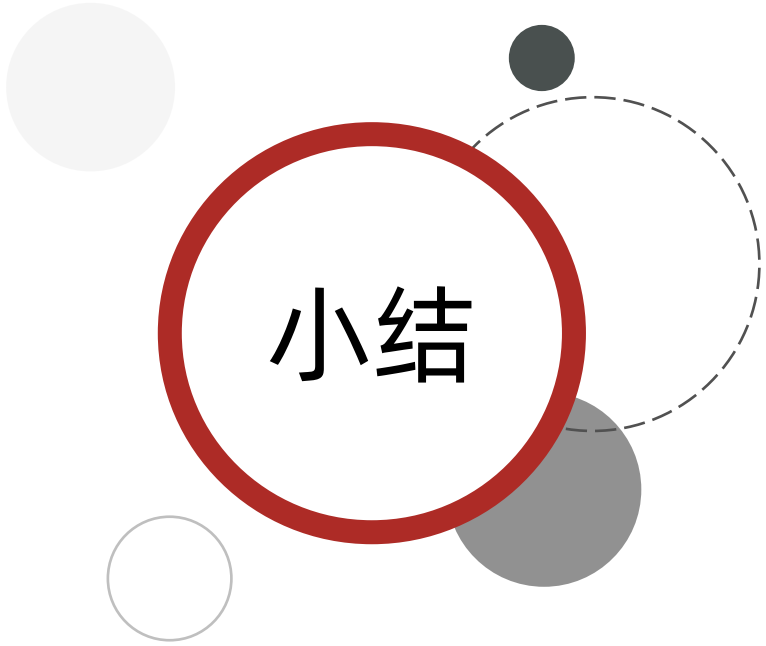
页面访问拦截：判断本地存储是否有token, 没有token拦截到登录

```
<!-- 本地存储有token，则说明用户登录了；没有token，则说明用户没有登录，不允许访问首页 -->
<script>
  if (localStorage.getItem('user-token') === null) location.href = './login.html'
</script>
```

思考：上述判断可否判断 token 的真假?

提示：前端只能判断有没有token，无法界定真假

后端是能知道真假的，所以将来我们请求后端后，需要对响应结果做判断 （后续会实现）



页面访问拦截的核心思路？

判断 token 是否存在

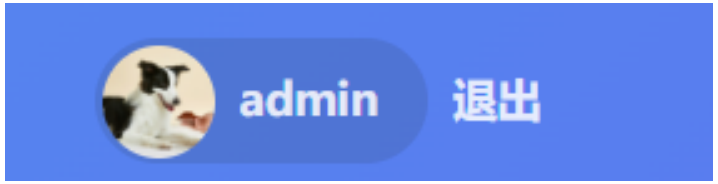
1. 有token，允许访问

2. 没有token，拦截到登录

需求：实现用户名的展示 和 退出登录

1. 用户名展示：读取本地数据，渲染视图

2. 退出登录： 注册事件，移除本地登录状态，跳转登录页




```
// 显示用户名称和退出登录
const userName = document.querySelector('.navbar .font-weight-bold')
const logout = document.querySelector('#logout')
if (userName) {
  userName.innerHTML = localStorage.getItem('user-name')
}
if (logout) {
  logout.addEventListener('click', () => {
    localStorage.removeItem('user-token')
    localStorage.removeItem('user-name')
    location.href = './login.html'
  })
}
```

目录

Contents

试一试：在首页获取后台统计数据？

```
// DOMContentLoaded 当初始的 HTML 文档被完全加载和解析完成之后,DOMContentLoaded 事件被触发
// 而无需等待样式表、图像和子框架的完成加载
document.addEventListener('DOMContentLoaded', async () => {
  const res = await axios.get('/dashboard')
  console.log(res);
})
```

发现401错误：

```
▶GET http://ajax-api.itheima.net/dashboard 401 (Unauthorized)        axios.js:328
▶Uncaught (in promise) Error: Request failed with status code 401    axios.js:864
    at createError (axios.js:864:15)
    at settle (axios.js:1160:12)
    at XMLHttpRequest.onloadend (axios.js:184:7)
```

原因？接口需要 请求头中 携带token

试一试：在首页获取后台统计数据？

```
// DOMContentLoaded 当初始的 HTML 文档被完全加载和解析完成之后,DOMContentLoaded 事件被触发
// 而无需等待样式表、图像和子框架的完成加载
document.addEventListener('DOMContentLoaded', async () => {
  const res = await axios.get('/dashboard')
  console.log(res);
})
```

请求头中携带token

```
document.addEventListener('DOMContentLoaded', async () => {
  const token = localStorage.getItem('user-token')
  const res = await axios.get('/dashboard', {
    headers: {
      'Authorization': token
    }
  })
  console.log(res);
})
```

思考

刚才我们请求后台接口时，为了让后台认识我们，

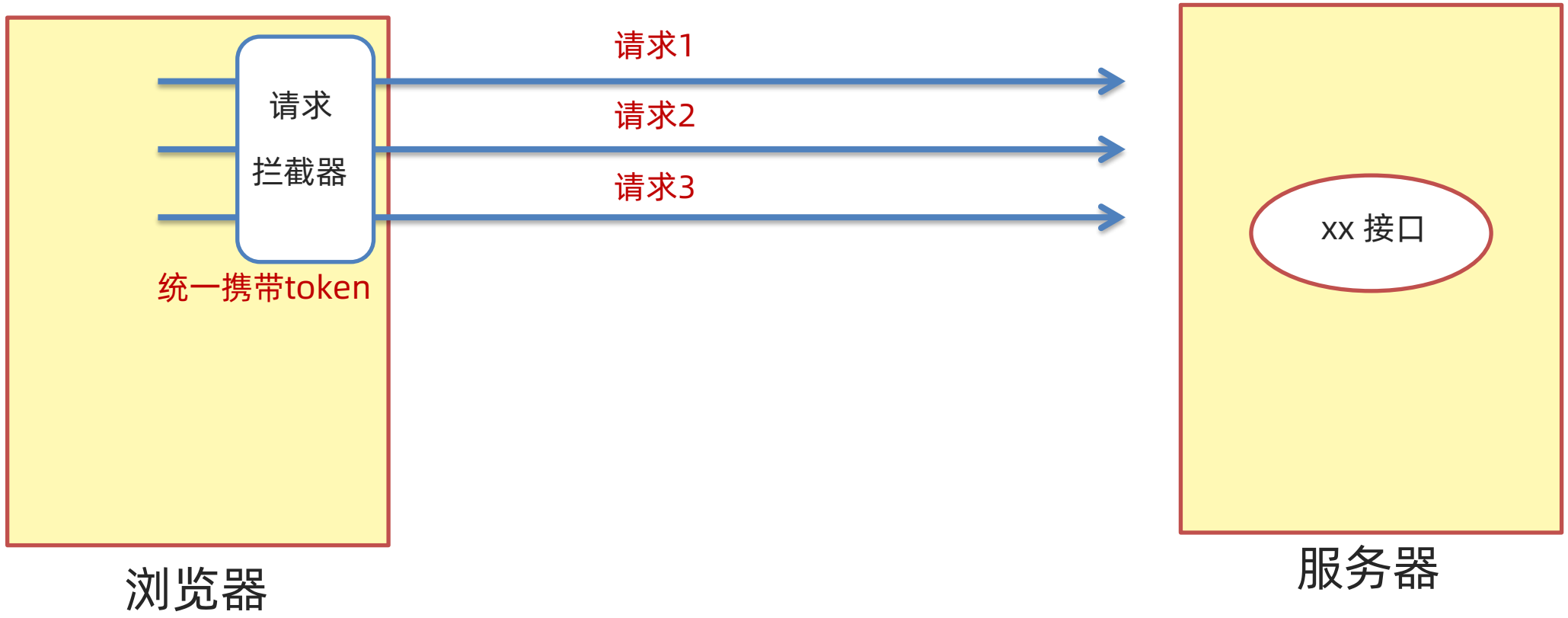
在请求头中携带了token

但其他接口是否也需要携带token呢？每次都请求时手动携带？

axios - 请求拦截器

axios请求拦截器：在所有axios请求 发送出去 之前，都会 先经过请求拦截器

应用场景：统一配置请求头令牌token。



axios - 请求拦截器

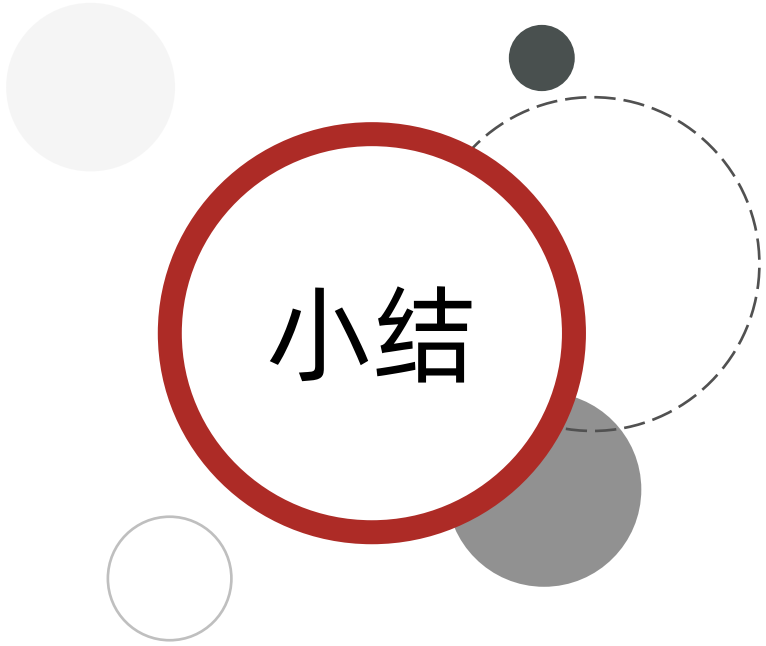
axios请求拦截器：在所有axios请求 发送出去 之前，都会 先经过请求拦截器

方案：请求拦截器，统一配置请求头令牌token。

```
// 添加请求拦截器
axios.interceptors.request.use(function (config) {

  // 在发送请求之前做些什么
  const token = localStorage.getItem('user-token')
  if (token) {
    config.headers.Authorization = token
  }

  return config;
}, function (error) {
  // 对请求错误做些什么
  return Promise.reject(error);
});
```

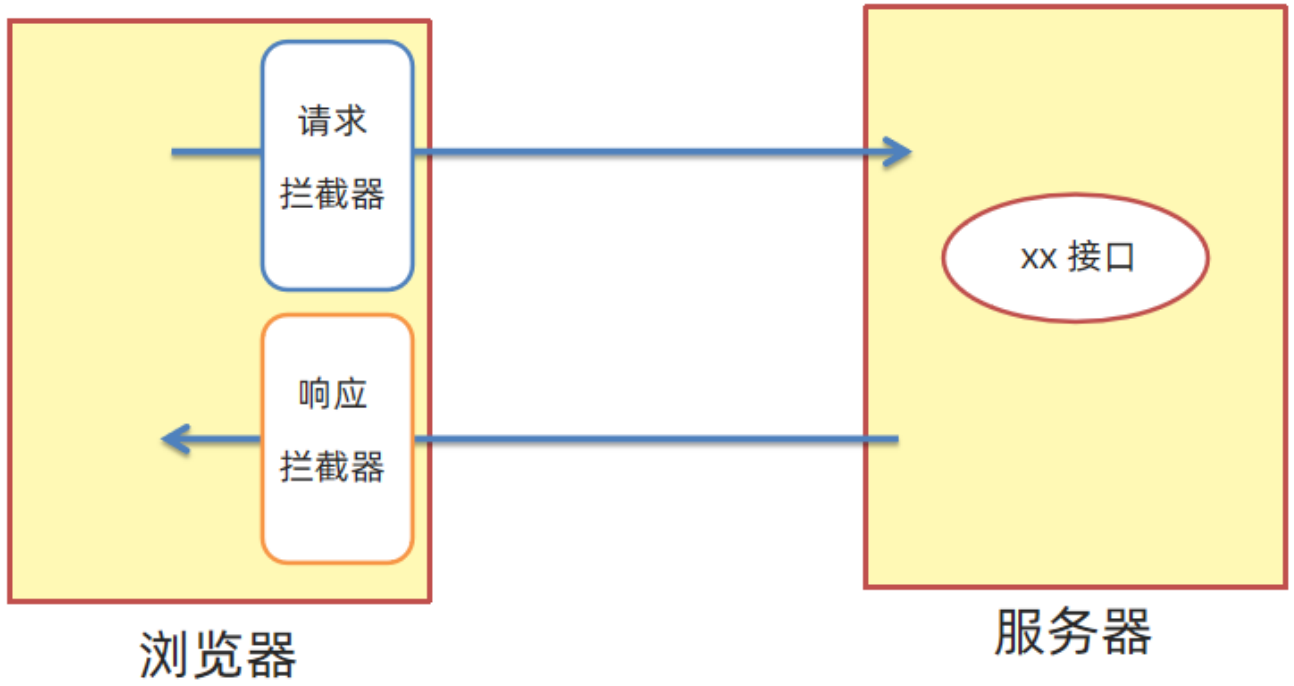


每个请求都要携带token，进行请求，如何操作？

利用请求拦截器，统一携带token

思考

刚才我们认识了请求拦截器，有没有响应拦截器呢？



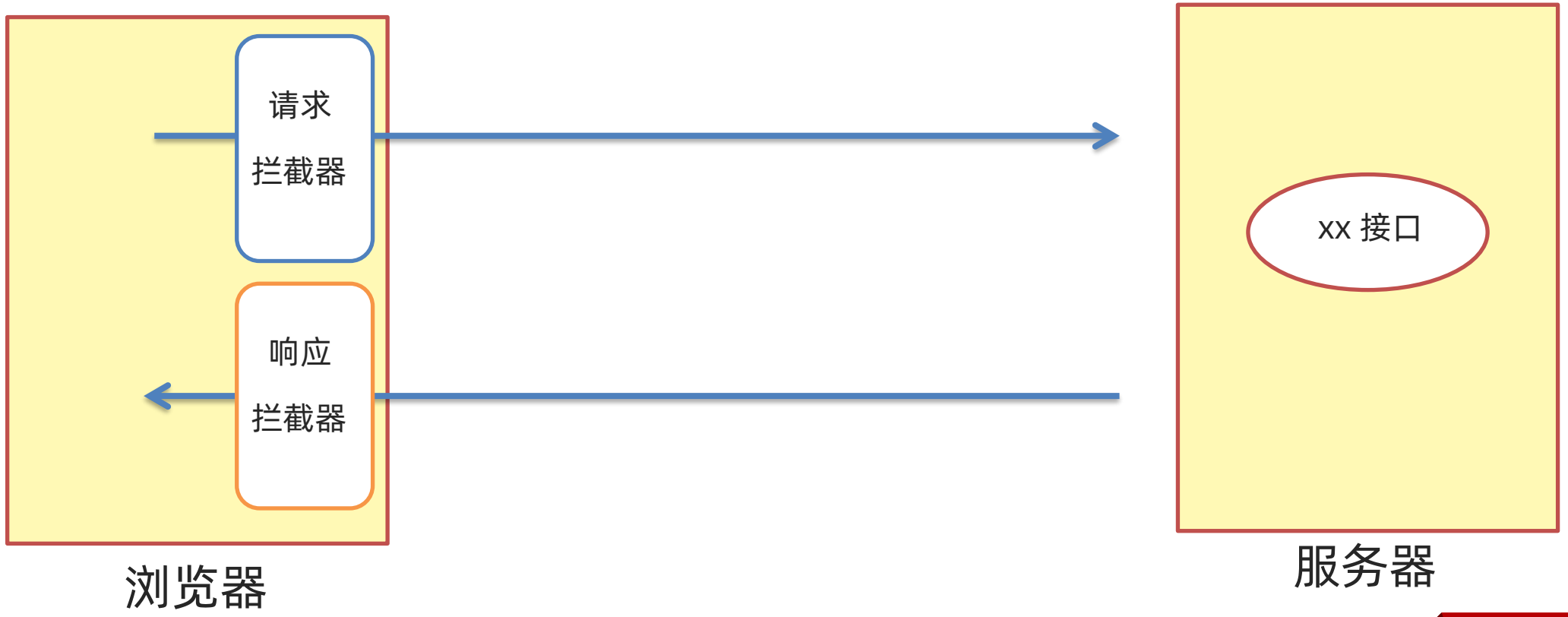
响应拦截器 一般用于 做什么呢？

axios - 响应拦截器

axios响应拦截器：在axios响应被.then/.catch处理之前，会先经过响应拦截器

应用场景：1. 统一处理错误响应（比如：token过期、 服务器错误）

2. 数据剥离 （去掉axios的默认一层data）

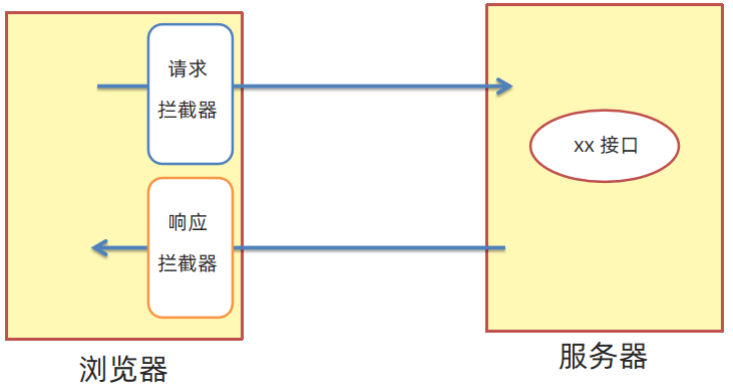


axios - 响应拦截器

axios响应拦截器：在axios响应被.then/.catch处理之前，会先经过响应拦截器

应用场景：1. 统一处理错误响应（比如：token过期、 服务器错误）

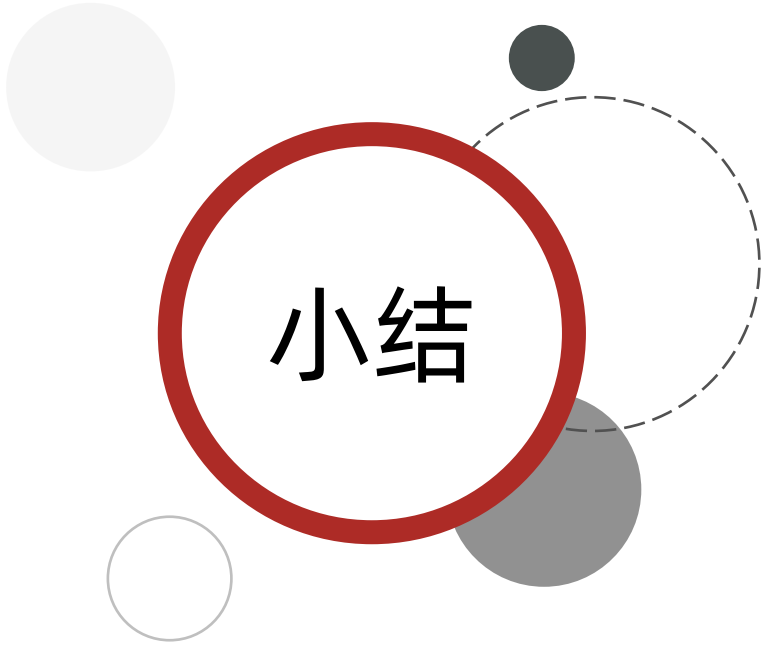
2. 数据剥离 （去掉axios的默认一层data）




```
// 添加响应拦截器
axios.interceptors.response.use(
  function (response) {
    // 对响应数据做点什么
    return response.data
  },
  function (error) {
    // 对响应错误做点什么
    if (error.response.status === 401) {
      localStorage.removeItem('user-token')
      localStorage.removeItem('user-name')
      location.href = './login.html'
    }
    return Promise.reject(error)
  }
)
```

数据剥离

处理响应错误



响应拦截器 常见使用场景 是什么呢？

1. 统一处理错误响应

2. 数据剥离，简化后台数据的访问层级

目录

Contents

我们已经开发了一定功能了，突然电脑坏了，怎么办？

功能完成，别人如何获取你写的代码？CV给他么？

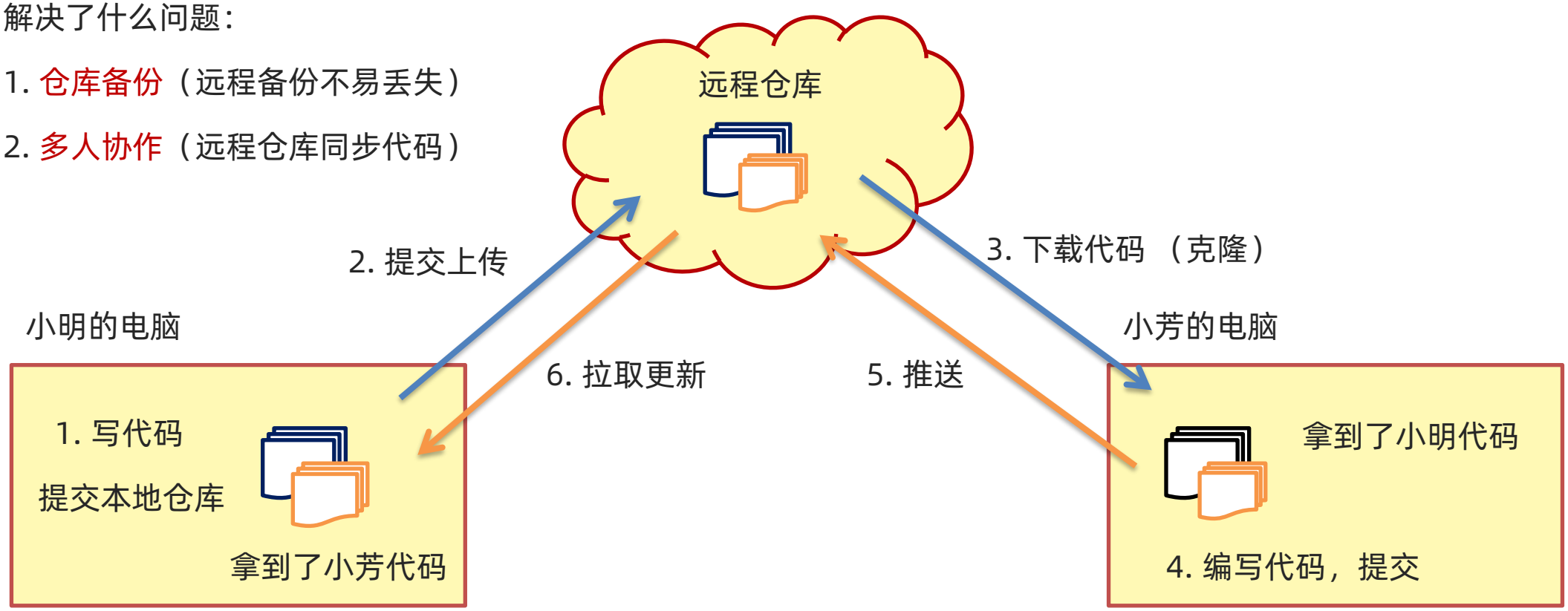
需要学习远程仓库！

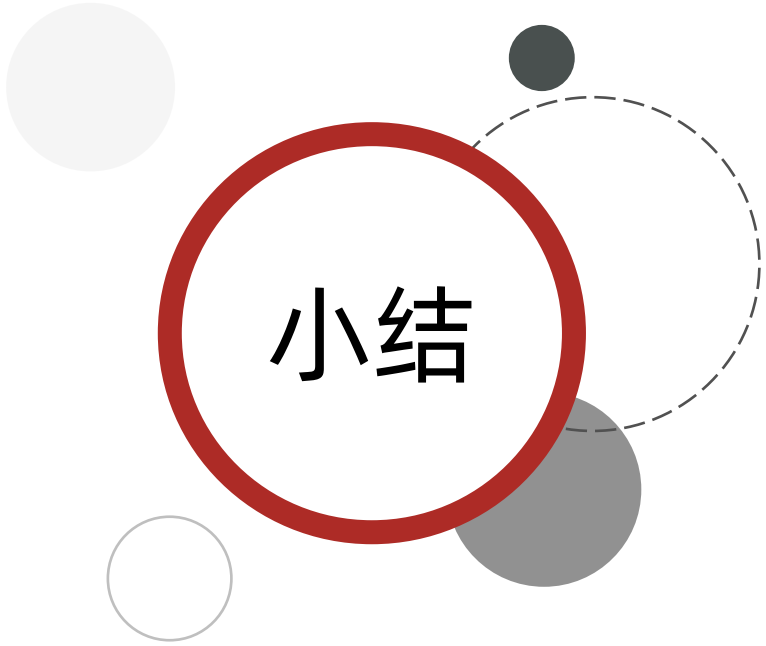
git 远程仓库的理解

远程仓库：托管在因特网或其他网络中，你的项目的版本库。支持多人协作，共同管理远程仓库。

解决了什么问题：

1. 仓库备份（远程备份不易丢失）
2. 多人协作（远程仓库同步代码）





远程仓库的存在解决了什么问题？

1. 仓库备份问题 （远程备份不易丢失）

2. 多人协作问题 （利用远程仓库同步代码）

如何创建一个远程仓库？在哪创建呢？

git代码托管平台

专门用于 存放远程仓库 的网站平台，就是代码托管平台

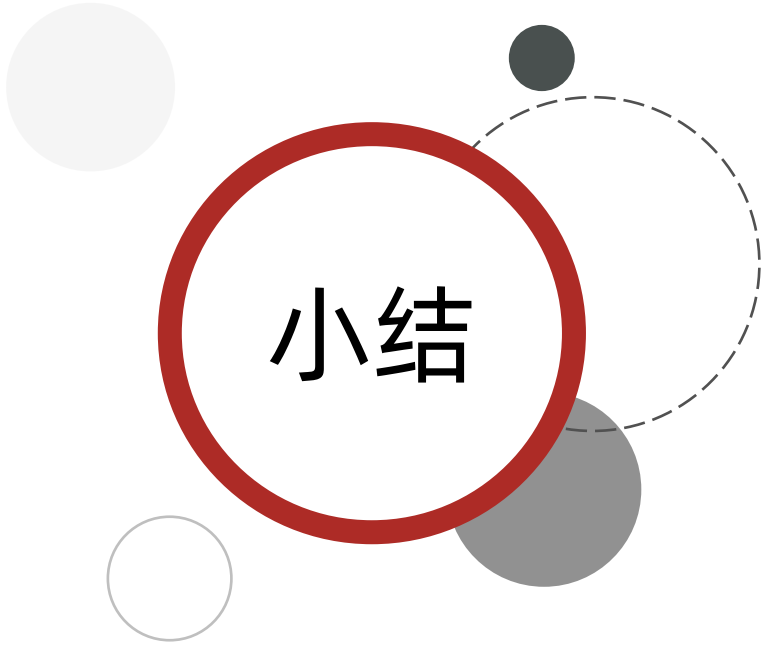
常见的代码托管平台：

- github 全球最大的开源项目托管平台 - 免费，国内访问受限，不稳定
- gitee 码云，国产开源项目托管平台。免费，访问速度快、纯中文界面、使用友好
- gitlab 对代码私有性支持较好，企业用户多，一般收费，允许自建 git 服务器

git代码托管平台

gitee 码云，创建远程仓库:

1. 新建仓库
2. 输入仓库名
3. 创建（不勾初始化！！！）



最常见的代码托管平台有几个？

3个。github gitee gitlab

不同的用户都在往平台上提交？

平台是如何区分认证不同的用户呢？

git支持多种传输协议：

最常用的两种传输协议：

- https协议：需要输入用户名和密码　　https://gitee.com/ruiweitang/test.git
- ssh协议：　需要配置秘钥，可免密码登录　　git@gitee.com:ruiweitang/test.git

- 注意：在实际公司开发中，ssh 的方式更为常见！更加安全可靠！

ssh秘钥的作用

ssh key 的**作用**：实现本地仓库和 gitee平台 之间免登录的加密数据传输。

ssh key 由**两部分组成**，分别是：

① id_rsa（私钥文件，存放于客户端的电脑中即可）

② id_rsa.pub（公钥文件，需要配置到 gitee平台 中）

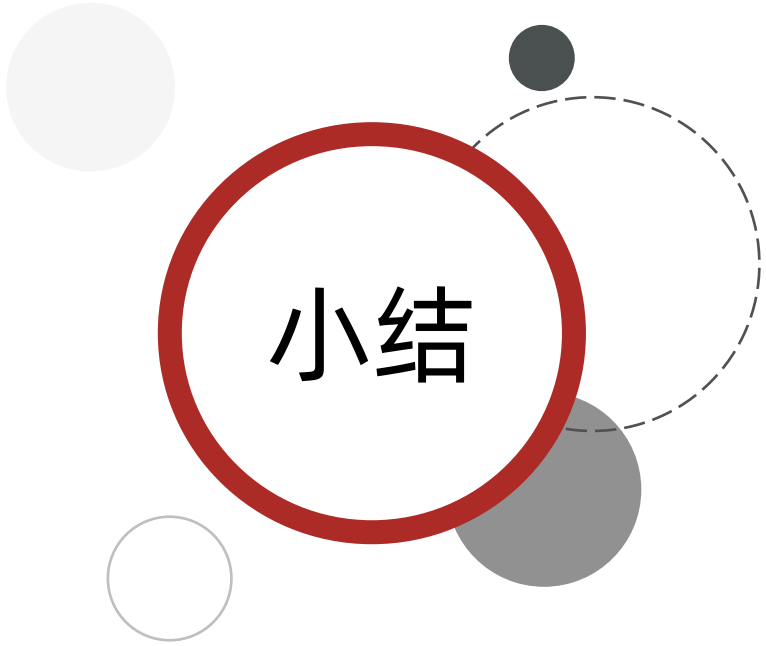
私钥加密的信息，只能通过公钥解密。公钥加密的信息，只能通过私钥解密。安全性高。

生成ssh key

步骤：

1. 打开 Git Bash
2. 粘贴如下的命令 ssh-keygen -t rsa
3. 连续敲击 3 次回车，即可在 C:\Users\用户名文件夹\.ssh 目录中生成 id_rsa 和 id_rsa.pub 两个文件
4. 使用记事本打开 id_rsa.pub 文件，复制里面的文本内容
5. 粘贴配置到 码云 -> 设置 -> ssh 公钥 中即可

tips: mac获取公钥 https://juejin.cn/post/6844904169191522317



我们刚才配置了什么？让gitee平台能够识别不同用户呢？

SSH密钥

本地已经有代码了，也有远程仓库了

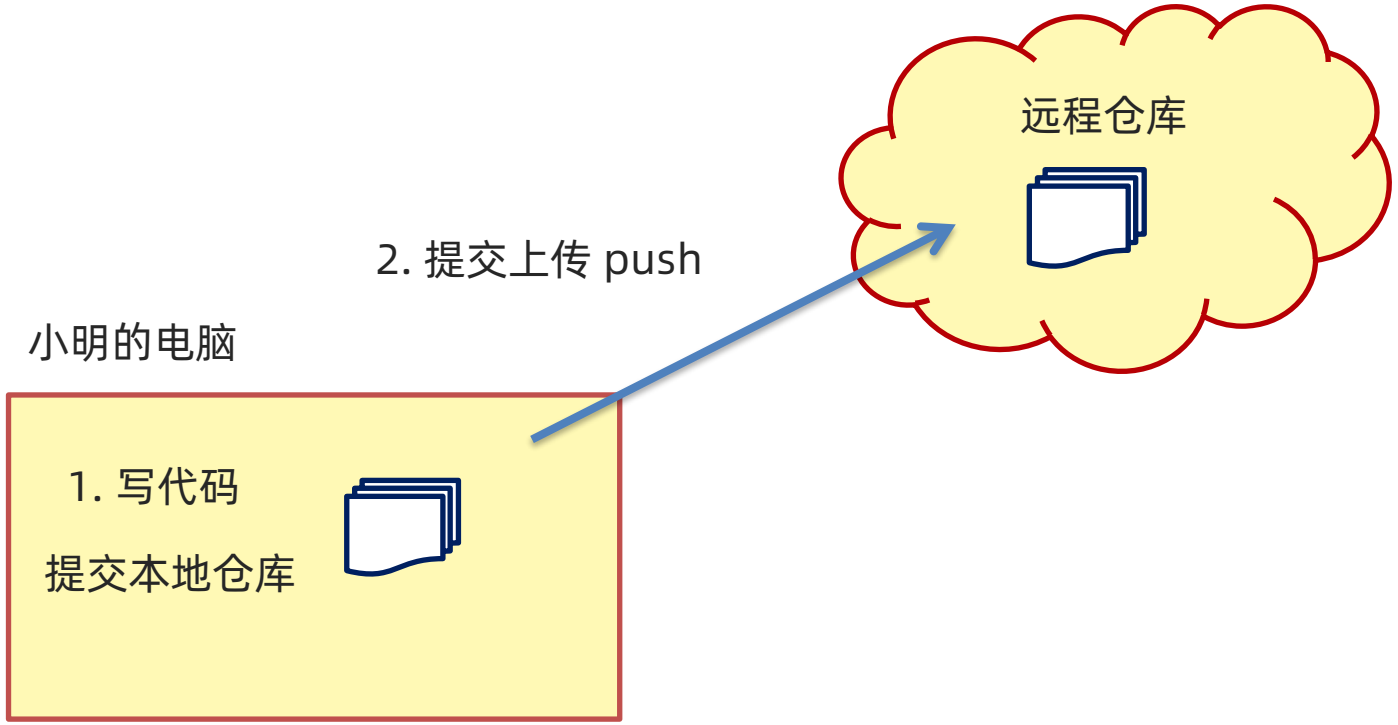
如何将本地仓库中的代码提交到远程仓库？

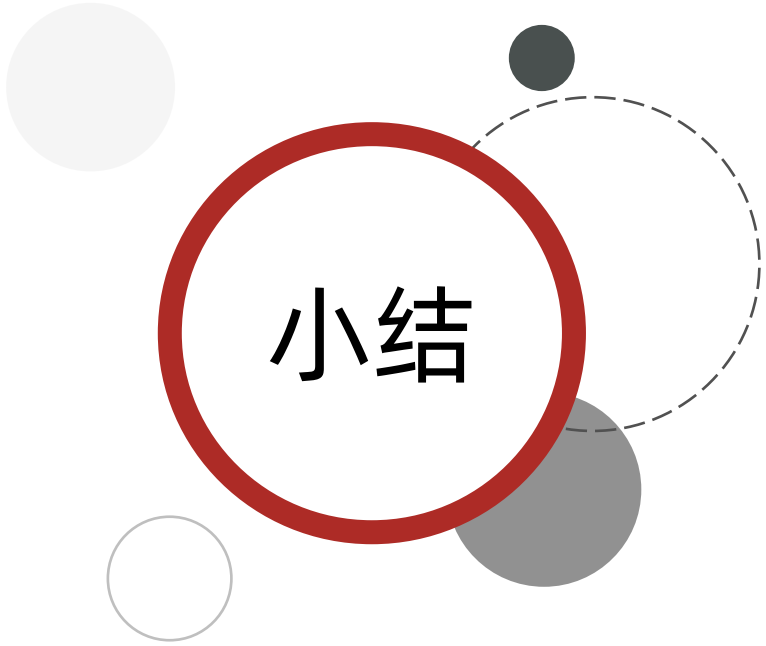
git push

作用：将本地仓库中代码提交到远程仓库

语法：git push 仓库地址 分支名

示例：git push git@gitee.com:ruiweitang/test.git master





如何将本地仓库中的代码提交到远程仓库？

git push 远程仓库地址 分支名

如果每次提交远程仓库，都要输入完整的仓库地址，麻烦么？

git push 远程仓库地址 分支名 （仓库地址不好记！！）

git remote

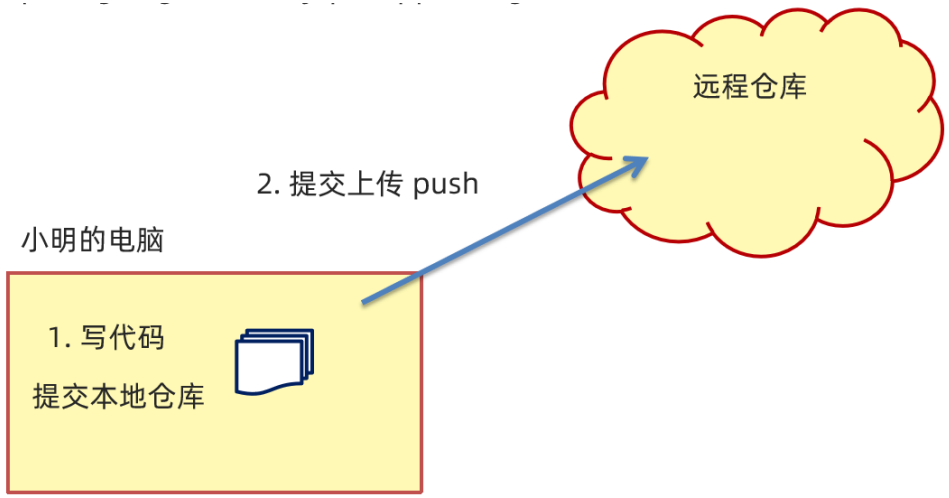
每次push操作都需要带上远程仓库的地址，非常的麻烦，我们可以给仓库地址设置一个别名

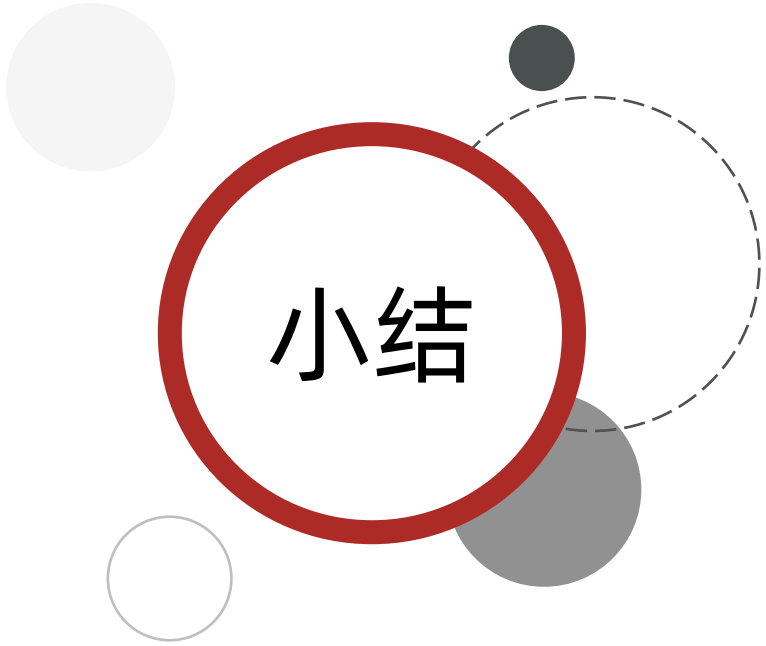
```
# 给远程仓库设置一个别名
git remote add 仓库别名 仓库地址
git remote add origin git@gitee.com:ruiweitang/test.git

# 删除origin这个别名
git remote remove origin
```

演示命令：git push -u 仓库别名 分支名

tips：-u 可以记录push到远端分支的默认值，将来 git push即可





如何设置远程仓库别名？如何删除？

添加：git remote add 别名 仓库地址

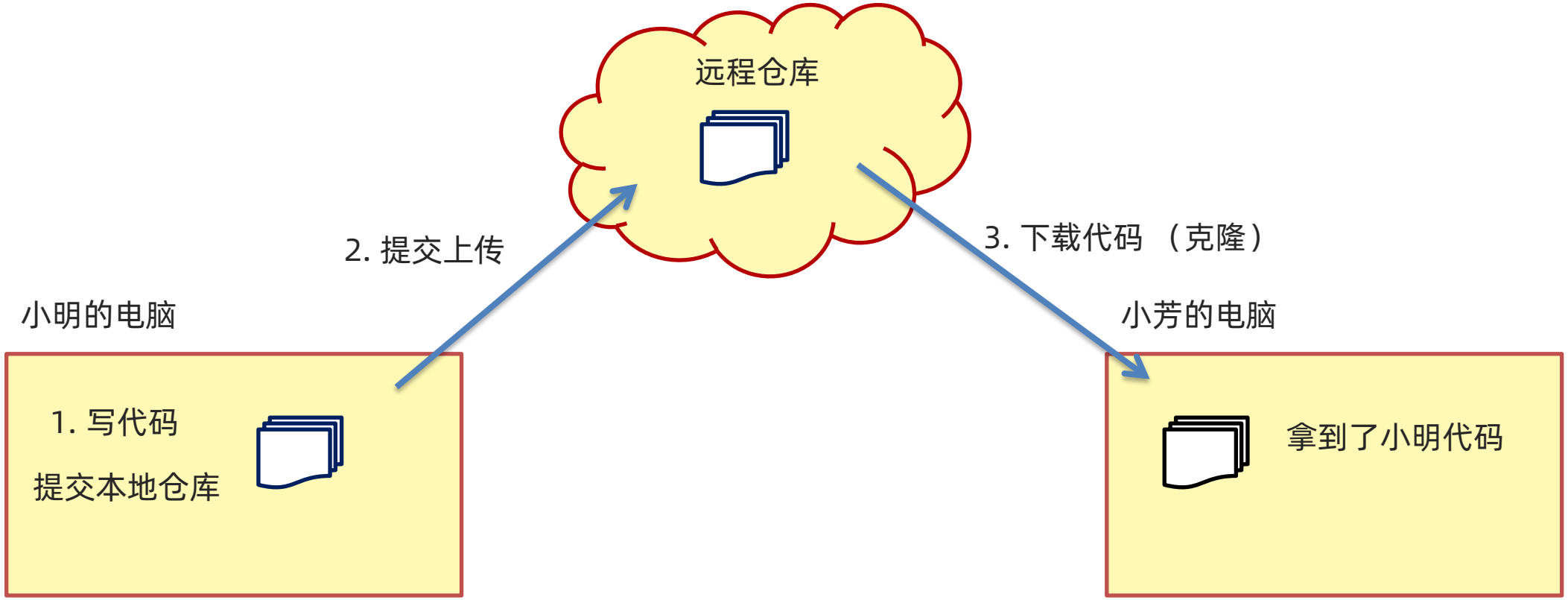
删除：git remote remove 别名

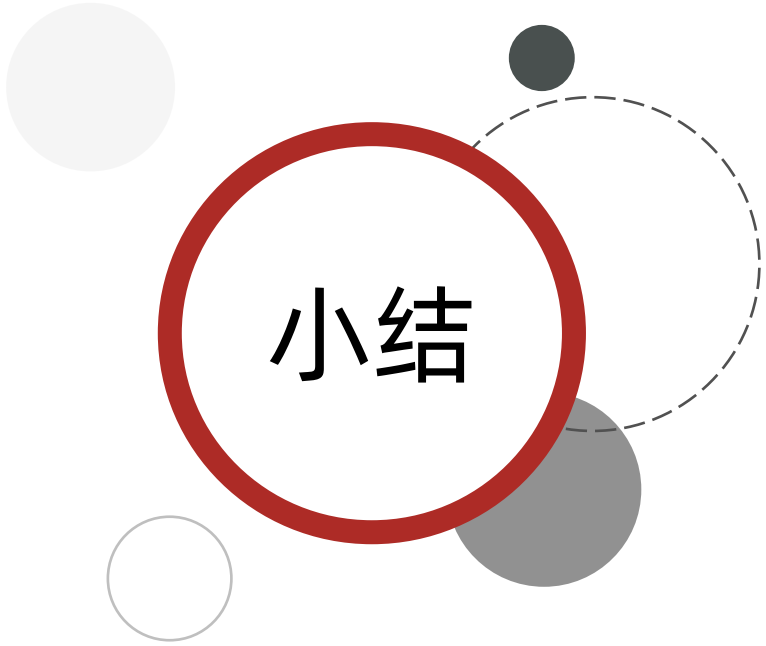
新入职一个小芳，如何拿到公司现有的代码呢？

git clone

作用：克隆远程仓库的代码到本地

语法：git clone [远程仓库地址] [文件夹名]





新入公司，如何获取公司源代码？

git clone 远程仓库地址

1. 拿到了源代码，完成了功能，又如何提交到远程仓库呢？ push

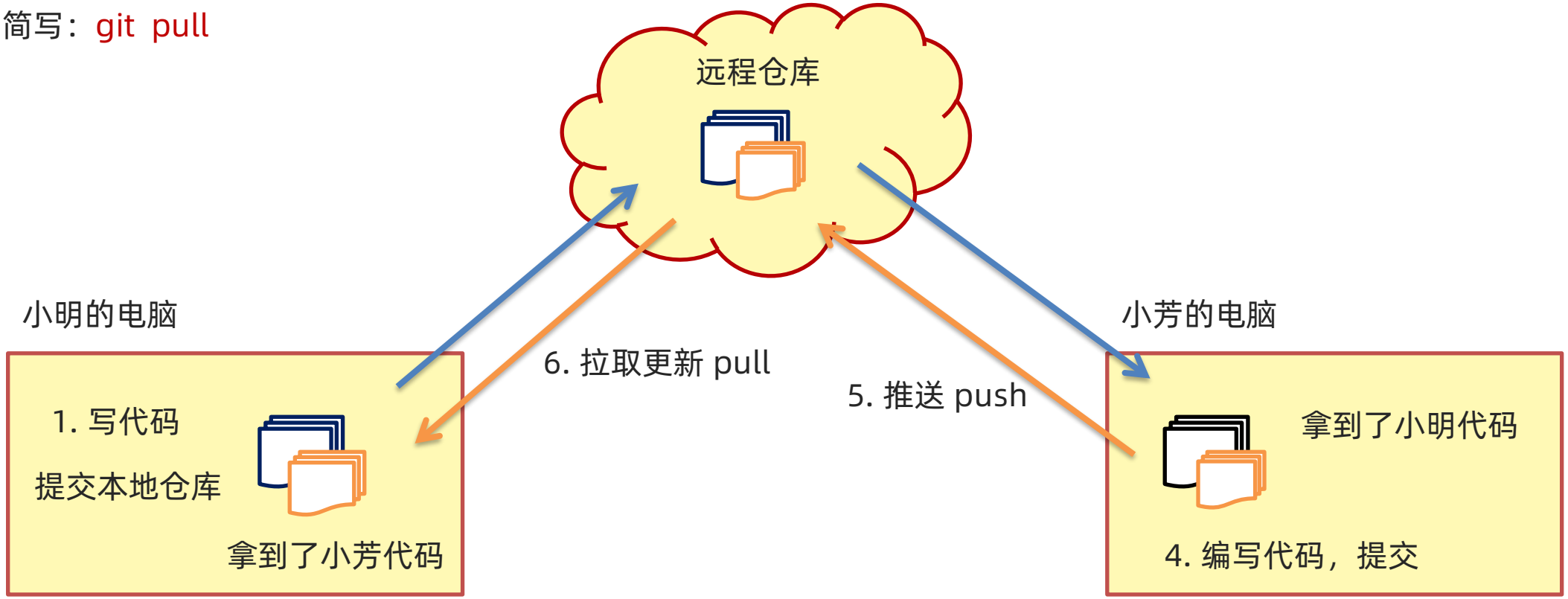
2. 提交到了远程仓库，别人又如何获取你的更新呢？

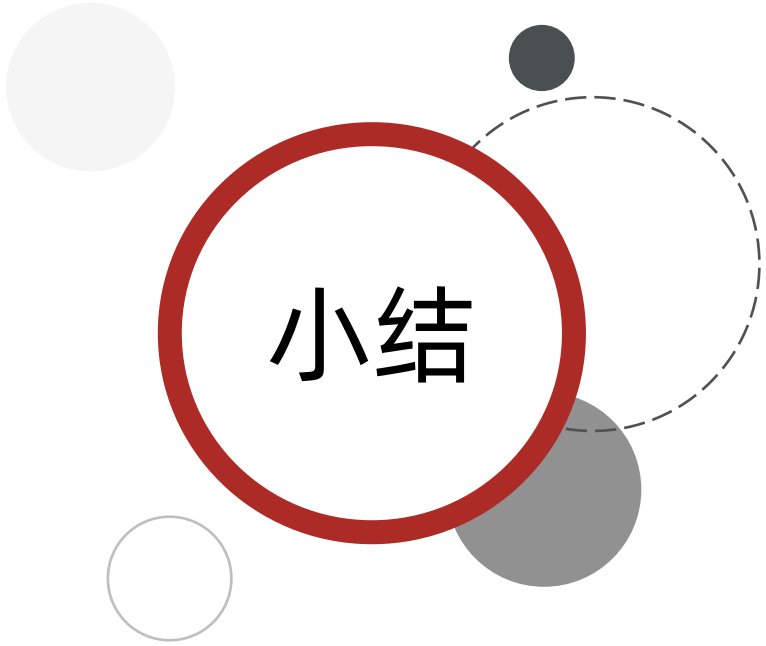
git pull

作用：拉取更新，将远程的代码下载合并到本地的分支

语法：git pull [远程仓库地址] [分支名]

简写：git pull





如何拉取别人代码的更新？

git pull origin 分支名

简写： git pull

如何下载别人的远端分支到本地？

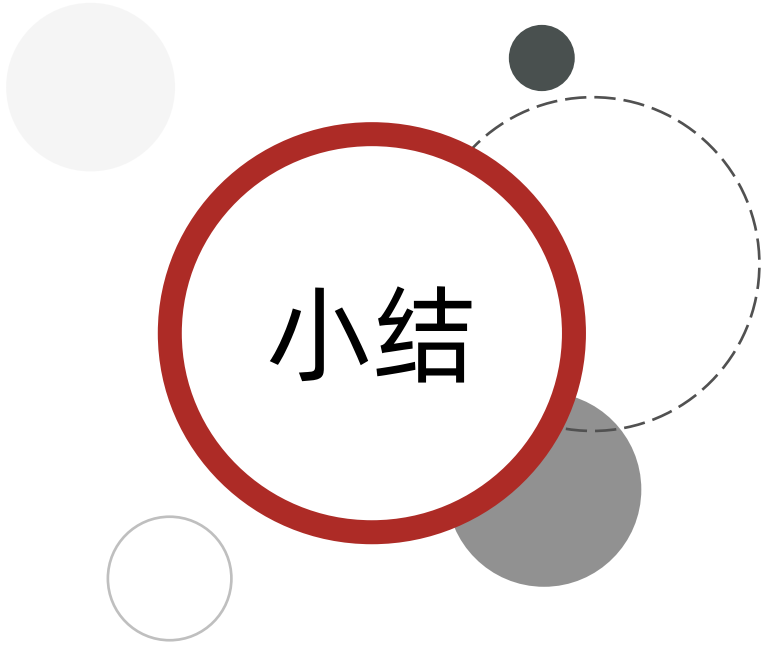
下载远端分支本地

命令：git checkout -t origin/分支名

作用：先在本地建立一个分支，并切换到该分支，然后从远程分支上同步代码到该分支上，并建立关联

后续拉取该分支的更新，就是切换到该分支，git pull origin 分支名

小技巧：这个命令不好记，可以利用vscode快速切换拉取



如何下载别人的远端分支到本地？

1. git checkout -t origin/分支名

2. vscode选择切换即可

综合练习 - 本地代码远程提交

需求1：将数据可视化项目上传到码云仓库，主分支与开发分支均要上传

参考命令：

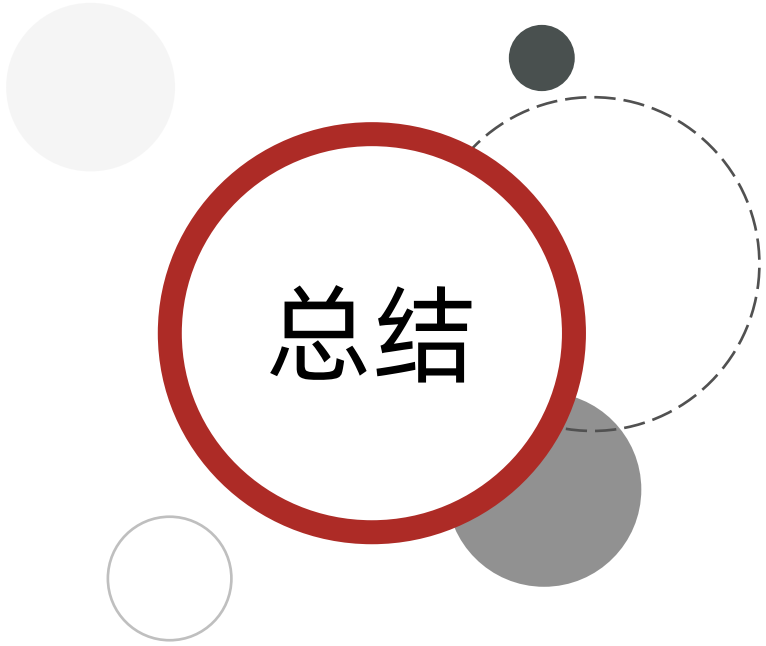
```
git  push  origin  master
git  checkout  develop
git  push  origin  develop
```

注意： 要推哪个分支，先切换到那个分支再推送

需求2：删除本地项目，通过克隆把远端仓库项目拉取下来，切换到develop分支

参考命令：

```
git  clone  远程仓库地址
git  checkout  -t  origin/develop
```



1. 学习 jwt 知识，实现页面访问权限控制
2. 能够掌握 axios 请求和响应拦截器的配置
3. 能够掌握 Git 远程仓库操作的命令
4. 综合练习：实现项目推送远程仓库

传智教育旗下高端IT教育品牌