使用token进行登录验证。所以需要做到发请求获取token，再通过vuex把token存储到本地。然后还要做到把token放到请求头，登录成功后还要获取用户信息,并且还需要新建一个用来展示用户信息的页面

登录涉及的文件有：

**api/login.js** 从登录接口获取请求还有获取用户登录信息接口获取请求

**mock/login.js** api向这里发请求获取假数据。造假登录接口

**views/login/index.vue** script里写请求获取token的请求函数，先进行表单验证。服务器响应返回token。登录成功后进行重定向跳转页面

**store/modules/app.js**  mutations和actions里面写存储token的方法setToken()，并引用utils/storage.js保存到locaStorage

**router/modules/login,js**  写登录页面的路由

**utils/request.js**  封装请求方法，包含请求与响应的拦截器，在请求拦截器中将token放到请求头

**utils/storage.js**  getItem和setItem是封装的操作localStorage的方法

api中import的是 request是因为文件名为requset。而不是因为导出request对象或方法。因为模块化要的就是文件名表明一个作用域。引用其他文件的内容也不是那么容易的，得带上文件名进行访问。当然api在引用request时还是和导出的对象息息相关的。相当于引用对象转换为导出对象进行一遍request文件流程。

登录获取用户信息的涉及文件有：

**permissions.js** 路由守卫，在上述路由跳转之前拿用户数据。白名单里的是不需要登录就可以访问的页面。写的是路由对象的name

**store/modules/account.js**  路由守卫使用其actions的getUserinfo方法向api/login.js发请求。并调用mutations的setUserinfo()将用户信息数据保存到vuex

**api/login.js** 获取用户登录信息接口

用户信息展示涉及文件：

**新建页面：**

1. 在`src/views`中创建一个`user`目录

2. 在`user`目录创建一个 index.vue，代表`用户管理`模块的第一个页面

3. 如果`用户管理`模块下还有其它页面，再创建`list.vue`

**创建路由：**

1. 在`src/router/modules`中创建一个`user.js`文件

2. 在`user.js`中配置路由

**创建api接口**

在`src/api`中创建一个`user.js`文件，该文件中包含了`用户管理`模块的所有接口

**准备mock数据**

新建mock/user.js

**views/user/index.vue** 页面发送请求获取用户数据