/\*相关依赖导入\*/

Procedure 登录模块

用户进入登录界面

用户输入账号密码

IF 用户点击登录（@click=login）

methods：{/\*方法调用login函数\*/

login(){

request.post([url:”/api/user/login”).then(res=>{](url:”/api/user/login”).then(res=%3e%7b)/\*向后台发送读取的用户账号密码数据\*/

if(res.code==’0’){/\*后台数据匹配成功\*/

this.$massage({

type:”success”

message:”登录成功”

}）

this.$router.push(“/”) //登录成功之后页面跳转到主页

}else{

this.$massage({

type:”error”

message：res.msg /\*返回登录错误的类型\*/

}）

}

})

}

}

ELSE

this.$router.push(“/login”)/\*返回到登录界面\*/

END IF

Procedure 用户注册模块

Import request from ‘@/utils/request’

用户填写相应数据

IF 用户点击注册（@click=submitForm(ruleform)）/\*\*/

  methods: {

    submitForm(formName) {/\*这是注册的函数，此处的formName中的数据应该是ruleForm,ruleForm中包含着数据库关于用户的字段，已经读取了用户输入的内容\*/

      this.$refs[formName].validate((valid) => {/\*校验，判断用户输入的数据是否满足条件\*/

        IF (valid) {

          this.loading = true/\*发出请求后一段时间不能再编辑，防止用户多次发出请求导致服务器报错\*/

          userRegister(this.ruleForm)/\*调用已经定义好的接口，将用户输入的表单字段发送到服务器 \*/

            .then((value) => {/\*value是服务器经过二次判断返回的json数据\*/

              const { code, message } = value

              if (code === 200) {/\*判断注册成功\*/

                this.$message({

                  message: '账号注册成功',

                  type: 'success'

                })

                setTimeout(() => {/\*跳转到登录界面\*/

                  this.loading = false

                  this.$router.push({ path: this.redirect || '/login' })

                }, 0.1 \* 1000)

              } else {

                this.$message.error('注册失败，' + message)

              }

            })

            .catch(() => {

              this.loading = false

            })

        } else {

          return false

        }

      })

    },

    resetForm(formName) {/\*清空表单\*/

      this.$refs[formName].resetFields()

    }

  }

ELSE

this.$router.push(“/register”)/\*返回到注册界面\*/

END IF