一、用户认证列表

1、用户认证列表接口

用户认证列表接口与用户列表一致，只是默认加了一个认证状态搜索条件：authStatus

2、用户认证列表前端

（1）添加路由

```
{
    path: 'userInfo/authList',
    name: '认证审批列表',
    component: () =>import('@/views/user/userInfo/authList'),
    meta: { title: '认证审批列表', icon: 'table' }
}
```

（2）添加用户认证列表页面

添加/views/user/userInfo/authList.vue组件

```
<template>
    <div class="app-container">
        <!--查询表单-->
        <el-form :inline="true" class="demo-form-inline">
            <el-form-item>
                <el-input v-model="searchObj.keyword" placeholder="姓名/手机"/>
            </el-form-item>

            <el-form-item label="创建时间">
                <el-date-picker
                    v-model="searchObj.createTimeBegin"
                    type="datetime"
                    placeholder="选择开始时间"
                    value-format="yyyy-MM-dd HH:mm:ss"
                    default-time="00:00:00"
                    />
            </el-form-item>
            至
            <el-form-item>
                <el-date-picker
                    v-model="searchObj.createTimeEnd"
                    type="datetime"
                    placeholder="选择截止时间"
                    value-format="yyyy-MM-dd HH:mm:ss"
                    default-time="00:00:00"
                    />
            </el-form-item>

            <el-button type="primary" icon="el-icon-search" @click="fetchData()">查询
            <el-button type="default" @click="resetData()">清空</el-button>
        </el-form>

        <!-- 列表 -->
        <el-table
        v-loading="listLoading"
        :data="list"
        stripe
            style="width: 100%">

            <el-table-column
            label="序号"
            width="70"
            align="center">
                <template slot-scope="scope">
                        {{ (page - 1) * limit + scope.$index + 1 }}
                </template>
            </el-table-column>

            <el-table-column prop="name" label="姓名"/>
            <el-table-column prop="certificatesType" label="证件类型"/>
            <el-table-column prop="certificatesNo" label="证件号"/>
            <el-table-column prop="createTime" label="创建时间"/>

            <el-table-column label="操作" width="250" align="center">
                <template slot-scope="scope">
                    <router-link :to="'/user/userInfo/show/'+scope.row.id">
```

```
                        <el-button type="primary" size="mini">查看</el-button>
                    </router-link>
                </template>
            </el-table-column>
        </el-table>

        <!-- 分页组件 -->
        <el-pagination
            :current-page="page"
            :total="total"
            :page-size="limit"
            :page-sizes="[5, 10, 20, 30, 40, 50, 100]"
            style="padding: 30px 0; text-align: center;"
            layout="sizes, prev, pager, next, jumper, ->, total, slot"
            @current-change="fetchData"
            @size-change="changeSize"
            />
    </div>
</template>
<script>
import userInfoApi from '@/api/userInfo'

export default {

// 定义数据
data() {
    return {
        listLoading: true, // 数据是否正在加载
        list: null, // banner列表
        total: 0, // 数据库中的总记录数
        page: 1, // 默认页码
        limit: 10, // 每页记录数
        searchObj: {
            authStatus: 1
        } // 查询表单对象
    }
},

// 当页面加载时获取数据
created() {
    this.fetchData()
},

methods: {
    // 调用api层获取数据库中的数据
    fetchData(page = 1) {
        console.log('翻页。。。' + page)
        // 异步获取远程数据（ajax）
        this.page = page
        userInfoApi.getPageList(this.page, this.limit, this.searchObj).then(
                response => {
            this.list = response.data.pageModel.records
            this.total = response.data.pageModel.total
            // 数据加载并绑定成功
            this.listLoading = false
        }
      )
    },

    // 当页码发生改变的时候
    changeSize(size) {
        console.log(size)
        this.limit = size
        this.fetchData(1)
    },

    // 重置查询表单
    resetData() {
        console.log('重置查询表单')
        this.searchObj = {}
        this.fetchData()
    }
  }
}
</script>
```

二、用户认证审批

1、用户认证审批接口

（1）添加UserInfoService接口

```
/**
```

```
 * 认证审批
 * @param userId
* @param authStatus 2：通过 -1：不通过
 */
void approval(Long userId, Integer authStatus);
```

(2）添加UserInfoServiceImpl实现

```
//认证审批  2通过  -1不通过
@Override
public void approval(Long userId, Integer authStatus) {
    if(authStatus.intValue()==2 || authStatus.intValue()==-1) {
        UserInfo userInfo = baseMapper.selectById(userId);
        userInfo.setAuthStatus(authStatus);
        baseMapper.updateById(userInfo);
    }
}
```

(3）添加UserController方法

```
//认证审批
@GetMapping("approval/{userId}/{authStatus}")
public R approval(@PathVariable Long userId,@PathVariable Integer authStatus) {
    userInfoService.approval(userId,authStatus);
    return R.ok();
}
```

2、用户认证审批前端

(1）封装api方法

```
  //认证审批
approval(id, authStatus) {
    return request({
      url: `${api_name}/approval/${id}/${authStatus}`,
      method: 'get'
    })
}
```

(2）用户认证列表页面添加审批按钮

```
<el-button type="primary" size="mini" @click="approval(scope.row.id, 2)">通过</el-but
<el-button type="danger" size="mini" @click="approval(scope.row.id, -1)">不通过</el-bu
```

(3）添加调用方法

```
// 审批
approval(id, authStatus) {
    // debugger
    this.$confirm('确定该操作吗?', '提示', {
        confirmButtonText: '确定',
        cancelButtonText: '取消',
        type: 'warning'
    }).then(() => { // promise
            // 点击确定，远程调用ajax
        return userInfoApi.approval(id, authStatus)
    }).then((response) => {
        this.fetchData(this.page)
        if (response.code) {
            this.$message({
                type: 'success',
                message: '操作成功!'
            })
        }
    })
},
```