**Open-his看病就诊系统前端开发笔记**

# 医生排班管理

## 测试数据

|  |
| --- |
| tableData: [ // 医生和科室信息mock数据，该数据假设已经按着部门信息进行排序          { userId: 2, deptId: 101, subsectionType: '1', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 2, deptId: 101, subsectionType: '2', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 2, deptId: 101, subsectionType: '3', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 3, deptId: 102, subsectionType: '1', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 3, deptId: 102, subsectionType: '2', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 3, deptId: 102, subsectionType: '3', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 4, deptId: 103, subsectionType: '1', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 4, deptId: 103, subsectionType: '2', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 4, deptId: 103, subsectionType: '3', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 5, deptId: 104, subsectionType: '1', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 5, deptId: 104, subsectionType: '2', schedulingType: ['1', '1', '1', '1', '1', '1', '1'] },          { userId: 5, deptId: 104, subsectionType: '3', schedulingType: ['', '1', '1', '1', '1', '1', '1'] }        ],        schedulingData: {          startTimeThisWeek: '2020-05-11',          endTimeThisWeek: '2020-05-17'        },        labelNames: [          '2020-07-27(周一)',          '2020-07-28(周二)',          '2020-07-29(周三)',          '2020-07-30(周四)',          '2020-07-31(周五)',          '2020-08-01(周六)',          '2020-08-02(周日)'        ], |

## 核心合并表格代码

|  |
| --- |
| spanMethod(data) { // 对于表格数据进行分组合并操作，UI组件回调函数        // const { row, column, rowIndex, columnIndex } = data        const { row, rowIndex, columnIndex } = data        if (columnIndex < 2 || columnIndex > 9) { // 医生合并展示区          const len = this.userGroup(row.userId)          const lenName = this.userIdLen(row.userId)          if (rowIndex === lenName) { // 某医生首位部门行            return {              rowspan: len,              colspan: 1            }          } else {            return { // 某医生非首位部门行              rowspan: 0,              colspan: 0            }          }        } else { // 排班信息展示区          return {            rowspan: 1,            colspan: 1          }        }      },      userGroup(name) { // 根据医生名称获取医生数量        return this.tableData.filter(o => o.userId === name).length      },      userIdLen(name) { // 根据医生名称获取该医生第一个部门在全量部门中的偏移位置        const tmp = Array.from(this.groupNum)        const index = tmp.indexOf(name) // 某医生在全医生中的偏移位置        let len = 0        for (let i = 0; i < index; i++) {          len += this.userGroup(tmp[i])        }        return len      }, |
| computed: {      groupNum() { // 获取医生列表数组        return new Set(this.tableData.map(o => o.userId))      }    }, |

## 创建src/api/doctor/scheduling.js

|  |
| --- |
| import request from '@/utils/request'  // 查询需要排除的用户列表 可以传部门ID 如果部门ID为空，则直接显示所有  export function queryUsersNeedScheduling(deptId) {    return request({      url: '/doctor/scheduling/queryUsersNeedScheduling/',      params: { deptId: deptId }    })  }  // 查询排班  export function queryScheduling(query) {    return request({      url: '/doctor/scheduling/queryScheduling',      method: 'get',      params: query    })  }  // 保存排班数据  export function saveScheduling(data) {    return request({      url: '/doctor/scheduling/saveScheduling',      method: 'post',      data: data    })  }  // 查询我的排班  export function queryMyScheduling(params) {    return request({      url: '/doctor/scheduling/queryMyScheduling',      method: 'get',      params: params    })  } |

## 修改src/routor/index.js



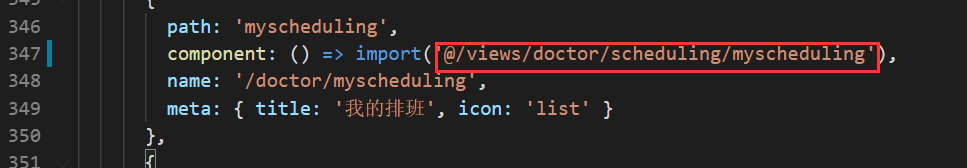
## 创建src/views/doctor/scheduling/index.vue静态

## 创建src/views/doctor/scheduling/index.vue最终动态

|  |
| --- |
| <template>    <div class="app-container">      <!-- 查询条件开始 -->      <el-row :gutter="12" style="margin-bottom:5px">        <el-col :span="24">          <el-card>            <el-form ref="queryForm" :model="queryParams" :inline="true" label-width="68px">              <el-form-item label="科室名称" prop="deptId">                <el-select                  v-model="queryParams.deptId"                  placeholder="请选择科室名称"                  clearable                  size="small"                  style="width:180px"                >                  <el-option                    v-for="d in deptOptions"                    :key="d.deptId"                    :label="d.deptName"                    :value="d.deptId"                  />                </el-select>              </el-form-item>              <el-form-item label="医生名称" prop="userId">                <el-select                  v-model="queryParams.userId"                  placeholder="请选择医生名称"                  clearable                  size="small"                  style="width:180px"                >                  <el-option                    v-for="d in userOptions"                    :key="d.userId"                    :label="d.userName"                    :value="d.userId"                  />                </el-select>              </el-form-item>              <el-form-item>                <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>                <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>              </el-form-item>              <el-form-item style="float:right">                <el-button type="primary" icon="el-icon-plus" size="small" @click="upWeek()">上一周</el-button>                <el-button type="success" icon="el-icon-s-operation" size="small" @click="currentWeek()">当前周</el-button>                <el-button type="primary" icon="el-icon-plus" size="small" @click="nextWeek()">下一周</el-button>              </el-form-item>            </el-form>          </el-card>        </el-col>      </el-row>      <!-- 查询条件结束 -->      <!-- 时间范围开始 -->      <el-row :gutter="12" style="margin-bottom:5px">        <el-col :span="24">          <el-card>            <div style="text-align:center">              <span>{{ schedulingData.startTimeThisWeek }}(周一)</span>-<span>{{ schedulingData.endTimeThisWeek }}(周日)</span>            </div>          </el-card>        </el-col>      </el-row>      <!-- 时间范围结束 -->      <!-- 排班表格开始 -->      <el-table v-loading="loading" :data="tableData" border :span-method="spanMethod">        <el-table-column prop="userId" label="医生" align="center" :formatter="userFormatter" />        <el-table-column prop="deptId" label="科室" align="center" :formatter="deptFormatter" />        <el-table-column prop="subsectionType" label="时间/日期" align="center" :formatter="subsectionTypeFormatter" />        <el-table-column prop="schedulingType[0]" :label="labelNames[0]" align="center" :formatter="schedulingTypeDay1Formatter" />        <el-table-column prop="schedulingType[1]" :label="labelNames[1]" align="center" :formatter="schedulingTypeDay2Formatter" />        <el-table-column prop="schedulingType[2]" :label="labelNames[2]" align="center" :formatter="schedulingTypeDay3Formatter" />        <el-table-column prop="schedulingType[3]" :label="labelNames[3]" align="center" :formatter="schedulingTypeDay4Formatter" />        <el-table-column prop="schedulingType[4]" :label="labelNames[4]" align="center" :formatter="schedulingTypeDay5Formatter" />        <el-table-column prop="schedulingType[5]" :label="labelNames[5]" align="center" :formatter="schedulingTypeDay6Formatter" />        <el-table-column prop="schedulingType[6]" :label="labelNames[6]" align="center" :formatter="schedulingTypeDay7Formatter" />        <el-table-column label="操作" align="center">          <template slot-scope="scope">            <el-button type="text" icon="el-icon-edit" size="mini" @click="editScheduling(scope.row.userId)">编辑</el-button>          </template>        </el-table-column>      </el-table>      <!-- 排班表格结束 -->      <!-- 修改的弹出层开始 -->      <el-dialog        :title="title"        :visible.sync="open"        width="1200px"        center        append-to-body      >        <el-table :data="editData" border>          <el-table-column prop="subsectionType" label="时间/日期" align="center" :formatter="subsectionTypeFormatter" />          <el-table-column prop="schedulingType[0]" :label="labelNames[0]" align="center">            <template slot-scope="scope">              <el-select                v-model="scope.row.schedulingType[0]"                clearable                size="small"              >                <el-option                  v-for="d in schedulingTypeOptions"                  :key="d.dictValue"                  :label="d.dictLabel"                  :value="d.dictValue"                />              </el-select>            </template>          </el-table-column>          <el-table-column prop="schedulingType[1]" :label="labelNames[0]" align="center">            <template slot-scope="scope">              <el-select                v-model="scope.row.schedulingType[1]"                clearable                size="small"              >                <el-option                  v-for="d in schedulingTypeOptions"                  :key="d.dictValue"                  :label="d.dictLabel"                  :value="d.dictValue"                />              </el-select>            </template>          </el-table-column>          <el-table-column prop="schedulingType[2]" :label="labelNames[0]" align="center">            <template slot-scope="scope">              <el-select                v-model="scope.row.schedulingType[2]"                clearable                size="small"              >                <el-option                  v-for="d in schedulingTypeOptions"                  :key="d.dictValue"                  :label="d.dictLabel"                  :value="d.dictValue"                />              </el-select>            </template>          </el-table-column>          <el-table-column prop="schedulingType[3]" :label="labelNames[0]" align="center">            <template slot-scope="scope">              <el-select                v-model="scope.row.schedulingType[3]"                clearable                size="small"              >                <el-option                  v-for="d in schedulingTypeOptions"                  :key="d.dictValue"                  :label="d.dictLabel"                  :value="d.dictValue"                />              </el-select>            </template>          </el-table-column>          <el-table-column prop="schedulingType[4]" :label="labelNames[0]" align="center">            <template slot-scope="scope">              <el-select                v-model="scope.row.schedulingType[4]"                clearable                size="small"              >                <el-option                  v-for="d in schedulingTypeOptions"                  :key="d.dictValue"                  :label="d.dictLabel"                  :value="d.dictValue"                />              </el-select>            </template>          </el-table-column>          <el-table-column prop="schedulingType[5]" :label="labelNames[0]" align="center">            <template slot-scope="scope">              <el-select                v-model="scope.row.schedulingType[5]"                clearable                size="small"              >                <el-option                  v-for="d in schedulingTypeOptions"                  :key="d.dictValue"                  :label="d.dictLabel"                  :value="d.dictValue"                />              </el-select>            </template>          </el-table-column>          <el-table-column prop="schedulingType[6]" :label="labelNames[0]" align="center">            <template slot-scope="scope">              <el-select                v-model="scope.row.schedulingType[6]"                clearable                size="small"              >                <el-option                  v-for="d in schedulingTypeOptions"                  :key="d.dictValue"                  :label="d.dictLabel"                  :value="d.dictValue"                />              </el-select>            </template>          </el-table-column>        </el-table>        <span slot="footer" class="dialog-footer">          <el-button type="primary" @click="handleSubmit">确 定</el-button>          <el-button @click="cancel">取 消</el-button>        </span>      </el-dialog>      <!-- 修改的弹出层结束 -->    </div>  </template>  <script>  import { queryUsersNeedScheduling, queryScheduling, saveScheduling } from '@/api/docter/scheduling'  import { selectAllDept } from '@/api/system/dept'  export default {    data() {      return {        // 遮罩层        loading: false,        // 科室数据        deptOptions: [],        // 用户数据        userOptions: [],        // 排班时间段字典        subsectionTypeOptions: [],        // 排班类型字典        schedulingTypeOptions: [],        // 查询条件        queryParams: {          deptId: undefined,          userId: undefined,          queryDate: undefined        },        schedulingData: {          startTimeThisWeek: undefined,          endTimeThisWeek: undefined        },        // 排班表的数据   应该从后台查询        tableData: [],        // 表头数据        labelNames: ['', '', '', '', '', '', ''],        // 弹出层的开启        open: false,        // 弹出层的标题        title: '',        // 弹出层里面的表格数据        editData: []      }    },    computed: {      groupNum() { // 获取医生列表数组        return new Set(this.tableData.map(o => o.userId))      }    },    created() {      // 查询科室      selectAllDept().then(res => {        this.deptOptions = res.data      })      // 查询要排班的医生      queryUsersNeedScheduling().then(res => {        this.userOptions = res.data      })      // 查询班类型      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 查询排班时间段      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 查询排班数据  返回值要填充schedulingData tableData labelNames      this.listScheduling()    },    methods: {      listScheduling() {        this.loading = true        // 调用API        queryScheduling(this.queryParams).then(res => {          console.log(res)          this.labelNames = res.data.labelNames          this.tableData = res.data.tableData          this.schedulingData = res.data.schedulingData          this.loading = false        })      },      // 查询      handleQuery() {        this.listScheduling()      },      // 重置      resetQuery() {        this.resetForm('queryForm')        this.listScheduling()      },      // 上一周      upWeek() {        this.queryParams.queryDate = this.schedulingData.startTimeThisWeek        this.listScheduling()      },      // 当前周      currentWeek() {        this.queryParams.queryDate = undefined        this.listScheduling()      },      // 下一周      nextWeek() {        this.queryParams.queryDate = this.schedulingData.endTimeThisWeek        this.listScheduling()      },      // 翻译用户      userFormatter(row) {        let name = ''        this.userOptions.filter(item => {          if (row.userId === item.userId) {            name = item.userName          }        })        return name      },      // 翻译科室      deptFormatter(row) {        let name = ''        this.deptOptions.filter(item => {          if (row.deptId === item.deptId) {            name = item.deptName          }        })        return name      },      // 翻译排班时段      subsectionTypeFormatter(row) {        return this.selectDictLabel(this.subsectionTypeOptions, row.subsectionType)      },      // 翻译周一排班类型      schedulingTypeDay1Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[0])      },      // 翻译周一排班类型      schedulingTypeDay2Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[1])      },      // 翻译周二排班类型      schedulingTypeDay3Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[2])      },      // 翻译周三排班类型      schedulingTypeDay4Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[3])      },      // 翻译周四排班类型      schedulingTypeDay5Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[4])      },      // 翻译周五排班类型      schedulingTypeDay6Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[5])      },      // 翻译周六排班类型      schedulingTypeDay7Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[6])      },      // 合并的代码      spanMethod(data) { // 对于表格数据进行分组合并操作，UI组件回调函数        const { row, rowIndex, columnIndex } = data        if (columnIndex < 2 || columnIndex > 9) { // 医生合并展示区          const len = this.userGroup(row.userId)          const lenName = this.userIdLen(row.userId)          if (rowIndex === lenName) { // 某医生首位部门行            return {              rowspan: len,              colspan: 1            }          } else {            return { // 某医生非首位部门行              rowspan: 0,              colspan: 0            }          }        } else { // 排班信息展示区          return {            rowspan: 1,            colspan: 1          }        }      },      userGroup(name) { // 根据医生名称获取医生数量        return this.tableData.filter(o => o.userId === name).length      },      userIdLen(name) { // 根据医生名称获取该医生第一个部门在全量部门中的偏移位置        const tmp = Array.from(this.groupNum)        const index = tmp.indexOf(name) // 某医生在全医生中的偏移位置        let len = 0        for (let i = 0; i < index; i++) {          len += this.userGroup(tmp[i])        }        return len      },      // 打开编辑页面      editScheduling(userId) {        this.editData = []        this.open = true        let name = ''        this.userOptions.filter(item => {          if (userId === item.userId) {            name = item.userName          }        })        this.title = '修改【' + name + '】的排班信息'        this.tableData.filter(item => {          if (item.userId === userId) {            this.editData.push(item)          }        })      },      // 关闭      cancel() {        this.open = false      },      // 保存      handleSubmit() {        const form = { beginDate: this.schedulingData.startTimeThisWeek, data: this.editData }        this.loading = true        // 保存        saveScheduling(form).then(res => {          // this.listScheduling()// 重新加载一次          this.loading = false          this.open = false          this.msgSuccess('保存成功')        }).catch(() => {          this.loading = false          this.msgError('保存失败')        })      }    }  }  </script> |

# 我的排班查询

## 修改src/routor/index.js



## 创建src/views/doctor/scheduling/myscheduling.vue

|  |
| --- |
| <template>    <div class="app-container">      <!-- 查询条件开始 -->      <el-row :gutter="12" style="margin-bottom:5px">        <el-col :span="24">          <el-card>            <el-form :inline="true" label-width="68px">              <el-form-item style="float:right">                <el-button type="primary" icon="el-icon-plus" size="small" @click="upWeek()">上一周</el-button>                <el-button type="success" icon="el-icon-s-operation" size="small" @click="currentWeek()">当前周</el-button>                <el-button type="primary" icon="el-icon-plus" size="small" @click="nextWeek()">下一周</el-button>              </el-form-item>            </el-form>          </el-card>        </el-col>      </el-row>      <!-- 查询条件结束 -->      <!-- 时间范围开始 -->      <el-row :gutter="12" style="margin-bottom:5px">        <el-col :span="24">          <el-card>            <div style="text-align:center">              <span>{{ schedulingData.startTimeThisWeek }}(周一)</span>-<span>{{ schedulingData.endTimeThisWeek }}(周日)</span>            </div>          </el-card>        </el-col>      </el-row>      <!-- 时间范围结束 -->      <!-- 排班表格开始 -->      <el-table v-loading="loading" :data="tableData" border :span-method="spanMethod">        <el-table-column prop="userId" label="医生" align="center" :formatter="userFormatter" />        <el-table-column prop="deptId" label="科室" align="center" :formatter="deptFormatter" />        <el-table-column prop="subsectionType" label="时间/日期" align="center" :formatter="subsectionTypeFormatter" />        <el-table-column prop="schedulingType[0]" :label="labelNames[0]" align="center" :formatter="schedulingTypeDay1Formatter" />        <el-table-column prop="schedulingType[1]" :label="labelNames[1]" align="center" :formatter="schedulingTypeDay2Formatter" />        <el-table-column prop="schedulingType[2]" :label="labelNames[2]" align="center" :formatter="schedulingTypeDay3Formatter" />        <el-table-column prop="schedulingType[3]" :label="labelNames[3]" align="center" :formatter="schedulingTypeDay4Formatter" />        <el-table-column prop="schedulingType[4]" :label="labelNames[4]" align="center" :formatter="schedulingTypeDay5Formatter" />        <el-table-column prop="schedulingType[5]" :label="labelNames[5]" align="center" :formatter="schedulingTypeDay6Formatter" />        <el-table-column prop="schedulingType[6]" :label="labelNames[6]" align="center" :formatter="schedulingTypeDay7Formatter" />      </el-table>      <!-- 排班表格结束 -->    </div>  </template>  <script>  import { queryUsersNeedScheduling, queryMyScheduling } from '@/api/docter/scheduling'  import { selectAllDept } from '@/api/system/dept'  export default {    data() {      return {        // 遮罩层        loading: false,        // 科室数据        deptOptions: [],        // 用户数据        userOptions: [],        // 排班时间段字典        subsectionTypeOptions: [],        // 排班类型字典        schedulingTypeOptions: [],        queryParams: {          queryDate: undefined        },        // 查询条件        schedulingData: {          startTimeThisWeek: undefined,          endTimeThisWeek: undefined        },        // 排班表的数据   应该从后台查询        tableData: [],        // 表头数据        labelNames: ['', '', '', '', '', '', '']      }    },    computed: {      groupNum() { // 获取医生列表数组        return new Set(this.tableData.map(o => o.userId))      }    },    created() {      // 查询科室      selectAllDept().then(res => {        this.deptOptions = res.data      })      // 查询要排班的医生      queryUsersNeedScheduling().then(res => {        this.userOptions = res.data      })      // 查询班类型      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 查询排班时间段      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 查询排班数据  返回值要填充schedulingData tableData labelNames      this.listScheduling()    },    methods: {      listScheduling() {        this.loading = true        // 调用API        queryMyScheduling(this.queryParams).then(res => {          this.labelNames = res.data.labelNames          this.tableData = res.data.tableData          this.schedulingData = res.data.schedulingData          this.loading = false        })      },      // 查询      handleQuery() {        this.listScheduling()      },      // 重置      resetQuery() {        this.resetForm('queryForm')        this.listScheduling()      },      // 上一周      upWeek() {        this.queryParams.queryDate = this.schedulingData.startTimeThisWeek        this.listScheduling()      },      // 当前周      currentWeek() {        this.queryParams.queryDate = undefined        this.listScheduling()      },      // 下一周      nextWeek() {        this.queryParams.queryDate = this.schedulingData.endTimeThisWeek        this.listScheduling()      },      // 翻译用户      userFormatter(row) {        let name = ''        this.userOptions.filter(item => {          if (row.userId === item.userId) {            name = item.userName          }        })        return name      },      // 翻译科室      deptFormatter(row) {        let name = ''        this.deptOptions.filter(item => {          if (row.deptId === item.deptId) {            name = item.deptName          }        })        return name      },      // 翻译排班时段      subsectionTypeFormatter(row) {        return this.selectDictLabel(this.subsectionTypeOptions, row.subsectionType)      },      // 翻译周一排班类型      schedulingTypeDay1Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[0])      },      // 翻译周一排班类型      schedulingTypeDay2Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[1])      },      // 翻译周二排班类型      schedulingTypeDay3Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[2])      },      // 翻译周三排班类型      schedulingTypeDay4Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[3])      },      // 翻译周四排班类型      schedulingTypeDay5Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[4])      },      // 翻译周五排班类型      schedulingTypeDay6Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[5])      },      // 翻译周六排班类型      schedulingTypeDay7Formatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType[6])      },      // 合并的代码      spanMethod(data) { // 对于表格数据进行分组合并操作，UI组件回调函数        const { row, rowIndex, columnIndex } = data        if (columnIndex < 2 || columnIndex > 9) { // 医生合并展示区          const len = this.userGroup(row.userId)          const lenName = this.userIdLen(row.userId)          if (rowIndex === lenName) { // 某医生首位部门行            return {              rowspan: len,              colspan: 1            }          } else {            return { // 某医生非首位部门行              rowspan: 0,              colspan: 0            }          }        } else { // 排班信息展示区          return {            rowspan: 1,            colspan: 1          }        }      },      userGroup(name) { // 根据医生名称获取医生数量        return this.tableData.filter(o => o.userId === name).length      },      userIdLen(name) { // 根据医生名称获取该医生第一个部门在全量部门中的偏移位置        const tmp = Array.from(this.groupNum)        const index = tmp.indexOf(name) // 某医生在全医生中的偏移位置        let len = 0        for (let i = 0; i < index; i++) {          len += this.userGroup(tmp[i])        }        return len      }    }  }  </script> |

# 患者管理

## 创建src/api/doctor/patient.js

|  |
| --- |
| import request from '@/utils/request'  // 分页查询所有患者信息  export function listPatientForPage(params) {    return request({      url: '/doctor/patient/listPatientForPage',      params: params    })  }  // 根据患者ID查询患者信息  export function getPatientById(patientId) {    return request({      url: '/doctor/patient/getPatientById/' + patientId,      method: 'get'    })  }  // 根据患者ID查询患者的档案信息  export function getPatientFileById(patientId) {    return request({      url: '/doctor/patient/getPatientFileById/' + patientId,      method: 'get'    })  } |

## 修改src/routor/index.js



## 创建src/views/doctor/patient/index.vue

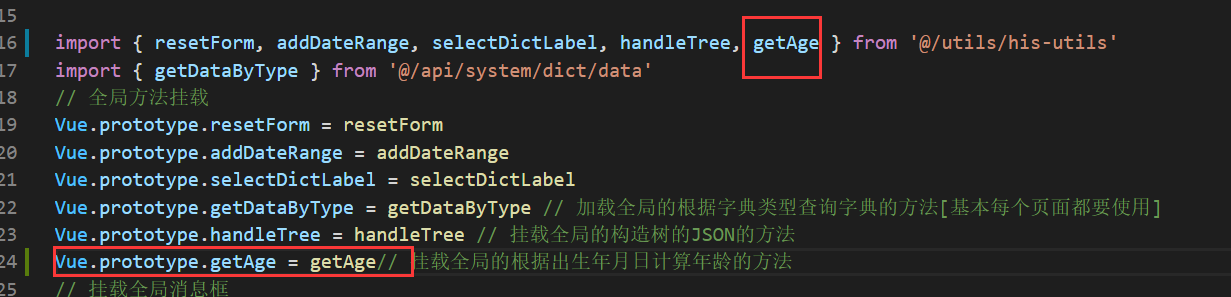
|  |
| --- |
| <template>    <div class="app-container">      <!-- 查询条件开始 -->      <el-form ref="queryForm" :model="queryParams" :inline="true" label-width="98px">        <el-form-item label="患者姓名" prop="name">          <el-input            v-model="queryParams.name"            placeholder="请输入患者姓名"            clearable            size="small"            style="width:180px"          />        </el-form-item>        <el-form-item label="手机号" prop="phone">          <el-input            v-model="queryParams.phone"            placeholder="请输入手机号"            clearable            size="small"            style="width:180px"          />        </el-form-item>        <el-form-item label="身份证号" prop="idCard">          <el-input            v-model="queryParams.idCard"            placeholder="请输身份证号"            clearable            size="small"            style="width:180px"          />        </el-form-item>        <el-form-item>          <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>          <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>        </el-form-item>      </el-form>      <!-- 查询条件结束 -->      <!-- 表格工具按钮开始 -->      <el-row :gutter="10" style="margin-bottom: 8px;">        <el-col :span="1.5">          <el-button type="primary" icon="el-icon-plus" size="mini" :disabled="single" @click="handleViewFile">查看档案</el-button>        </el-col>        <el-col :span="1.5">          <el-button type="success" icon="el-icon-edit" size="mini" :disabled="single" @click="handleViewCareHistory">查看就诊病例</el-button>        </el-col>      </el-row>      <!-- 表格工具按钮结束 -->      <!-- 数据表格开始 -->      <el-table v-loading="loading" border :data="patientTableList" @selection-change="handleSelectionChange">        <el-table-column type="selection" width="55" align="center" />        <el-table-column label="姓名" align="center" prop="name" />        <el-table-column label="电话" align="center" prop="phone" />        <el-table-column label="身份证号" align="center" prop="idCard" />        <el-table-column label="出生年月" align="center" prop="birthday">          <template slot-scope="scope">            <span>{{ scope.row.birthday.substring(0,10) }}</span>          </template>        </el-table-column>        <el-table-column label="年龄" align="center" prop="phone" :formatter="calculateAgeFormatter" />        <el-table-column label="性别" prop="sex" align="center" :formatter="sexFormatter" />        <el-table-column label="信息状态" prop="isFinal" align="center" :formatter="isFinalFormatter" />        <el-table-column label="创建时间" align="center" prop="createTime" />      </el-table>      <!-- 数据表格结束 -->      <!-- 分页控件开始 -->      <el-pagination        v-show="total>0"        :current-page="queryParams.pageNum"        :page-sizes="[5, 10, 20, 30]"        :page-size="queryParams.pageSize"        layout="total, sizes, prev, pager, next, jumper"        :total="total"        @size-change="handleSizeChange"        @current-change="handleCurrentChange"      />      <!-- 分页控件结束 -->      <!-- 查看患者档案弹出层开始 -->      <el-dialog        :title="title"        :visible.sync="fileOpen"        width="1000px"        center        append-to-body      >        <el-form label-position="left" label-width="120px" inline class="demo-table-expand">          <el-form-item label="ID:">            <span v-text="patientObj.patientId" />          </el-form-item>          <el-form-item label="姓名:">            <span v-text="patientObj.name" />          </el-form-item>          <el-form-item label="身份证号:">            <span v-text="patientObj.idCard" />          </el-form-item>          <el-form-item label="电话:">            <span v-text="patientObj.phone" />          </el-form-item>          <el-form-item label="患者性别:">            <span v-text="patientObj.sex" />          </el-form-item>          <el-form-item label="出生年月:">            <span v-text="patientObj.birthday" />          </el-form-item>          <el-form-item label="地址信息:">            <span v-text="patientObj.address" />          </el-form-item>          <el-form-item label="过敏信息:">            <span v-text="patientObj.allergyInfo" />          </el-form-item>          <el-form-item label="是否完善信息:">            <span v-text="patientObj.isFinal" />          </el-form-item>          <el-form-item label="最后登录ip:">            <span v-text="patientObj.lastLoginIp" />          </el-form-item>          <el-form-item label="最后登录时间:">            <span v-text="patientObj.lastLoginTime" />          </el-form-item>          <el-form-item label="创建时间:">            <span v-text="patientObj.createTime" />          </el-form-item>          <el-form-item label="更新时间:">            <span v-text="patientObj.updateTime" />          </el-form-item>          <el-form-item label="紧急联系人:">            <span v-text="patientFileObj.emergencyContactName" />          </el-form-item>          <el-form-item label="紧急联系人电话:">            <span v-text="patientFileObj.emergencyContactPhone" />          </el-form-item>          <el-form-item label="关系:">            <span v-text="patientFileObj.emergencyContactRelation" />          </el-form-item>          <el-form-item label="左耳听力:">            <span v-text="patientFileObj.leftEarHearing" />          </el-form-item>          <el-form-item label="右耳听力:">            <span v-text="patientFileObj.rightEarHearing" />          </el-form-item>          <el-form-item label="左眼视力:">            <span v-text="patientFileObj.leftVision" />          </el-form-item>          <el-form-item label="右眼视力:">            <span v-text="patientFileObj.rightVision" />          </el-form-item>          <el-form-item label="身高:">            <span v-text="patientFileObj.height" />          </el-form-item>          <el-form-item label="体重:">            <span v-text="patientFileObj.weight" />          </el-form-item>          <el-form-item label="血型:">            <span v-text="patientFileObj.bloodType" />          </el-form-item>          <el-form-item label="个人史:">            <span v-text="patientFileObj.personalInfo" />          </el-form-item>          <el-form-item label="家族史:">            <span v-text="patientFileObj.familyInfo" />          </el-form-item>          <el-form-item label="档案创建时间:">            <span v-text="patientFileObj.createTime" />          </el-form-item>          <el-form-item label="档案更新时间:">            <span v-text="patientFileObj.updateTime" />          </el-form-item>        </el-form>        <span slot="footer" class="dialog-footer">          <el-button @click="cancelFile">取 消</el-button>        </span>      </el-dialog>      <!-- 查看患者档案弹出层结束 -->    </div>  </template>  <script>  // 引入api  import { listPatientForPage, getPatientById, getPatientFileById } from '@/api/docter/patient'  export default {    // 定义页面数据    data() {      return {        // 是否启用遮罩层        loading: false,        // 选中数组        ids: [],        // 非单个禁用        single: true,        // 非多个禁用        multiple: true,        // 分页数据总条数        total: 0,        // 字典表格数据        patientTableList: [],        // 弹出层标题        title: '',        // 是否显示档案弹出层        fileOpen: false,        // 性别数据字典        sexOptions: [],        // 完善信息状态        isFinalOptions: [],        // 查询参数        queryParams: {          pageNum: 1,          pageSize: 10,          name: undefined,          idCard: undefined,          phone: undefined        },        // 患者信息        patientObj: {},        // 档案信息        patientFileObj: {}      }    },    // 勾子    created() {      // 加载性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })      // 加载性别      this.getDataByType('his\_patient\_msg\_final').then(res => {        this.isFinalOptions = res.data      })      // 查询表格数据      this.getPatientList()    },    // 方法    methods: {      // 查询表格数据      getPatientList() {        this.loading = true // 打开遮罩        listPatientForPage(this.addDateRange(this.queryParams, this.dateRange)).then(res => {          this.patientTableList = res.data          this.total = res.total          this.loading = false// 关闭遮罩        })      },      // 条件查询      handleQuery() {        this.getPatientList()      },      // 重置查询条件      resetQuery() {        this.resetForm('queryForm')        this.getPatientList()      },      // 数据表格的多选择框选择时触发      handleSelectionChange(selection) {        this.ids = selection.map(item => item.patientId)        this.single = selection.length !== 1        this.multiple = !selection.length      },      // 分页pageSize变化时触发      handleSizeChange(val) {        this.queryParams.pageSize = val        // 重新查询        this.getPatientList()      },      // 点击上一页  下一页，跳转到哪一页面时触发      handleCurrentChange(val) {        this.queryParams.pageNum = val        // 重新查询        this.getPatientList()      },      // 翻译性别      sexFormatter(row) {        return this.selectDictLabel(this.sexOptions, row.sex)      },      // 翻译信息状态      isFinalFormatter(row) {        return this.selectDictLabel(this.isFinalOptions, row.isFinal)      },      // 根据出生年月计算年龄      calculateAgeFormatter(row) {        return this.getAge(row.birthday.substring(0, 10))      },      // 查询患者档案      handleViewFile() {        const patientId = this.ids[0]        this.fileOpen = true        this.title = '查询患者档案信息'        this.patientFileObj = {}        this.patientObj = {}        // 查询患者        getPatientById(patientId).then(res => {          this.patientObj = res.data          const sex = this.patientObj.sex          const isFinal = this.patientObj.isFinal          this.patientObj.sex = (sex === '0' ? '男' : sex === '1' ? '女' : '未知')          this.patientObj.isFinal = isFinal === '0' ? '未完善' : '已完善'        })        // 查询档案        getPatientFileById(patientId).then(res => {          if (res.data !== null) {            this.patientFileObj = res.data          }        })        //      },      // 取消      cancelFile() {        this.fileOpen = false        this.title = ''      },      // 查询患者就诊信息      handleViewCareHistory(row) {      }    }  }  </script>  <style scoped>    .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }  </style> |

## 解决年龄显示的问题

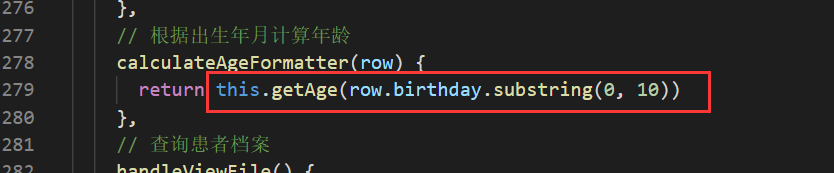
### 修改src/utils/his-utils

|  |
| --- |
| /\*\*   \* 公共的根据出生年月计算年龄的方法   \*  @param birthday 格式必须为2020-08-08  \*/  export function getAge(birthday) {    var birArr = birthday.split('-')    var birYear = parseInt(birArr[0])    var birMonth = parseInt(birArr[1])    var birDay = parseInt(birArr[2])    d = new Date()    var nowYear = d.getFullYear()    var nowMonth = d.getMonth() + 1 // 记得加1    var nowDay = d.getDate()    var returnAge    if (birArr == null) {      return false    }    var d = new Date(birYear, birMonth - 1, birDay)    if (d.getFullYear() === birYear && (d.getMonth() + 1) === birMonth && d.getDate() === birDay) {      if (nowYear === birYear) {        returnAge = 0 //      } else {        var ageDiff = nowYear - birYear //        if (ageDiff > 0) {          if (nowMonth === birMonth) {            var dayDiff = nowDay - birDay //            if (dayDiff < 0) {              returnAge = ageDiff - 1            } else {              returnAge = ageDiff            }          } else {            var monthDiff = nowMonth - birMonth //            if (monthDiff < 0) {              returnAge = ageDiff - 1            } else {              returnAge = ageDiff            }          }        } else {          return '出生日期晚于今天，数据有误' // 返回-1 表示出生日期输入错误 晚于今天        }      }      return returnAge    } else {      return ('输入的日期格式错误！')    }  } |

### 修改src/main.js



### 修改src/views/doctor/patient/index.vue

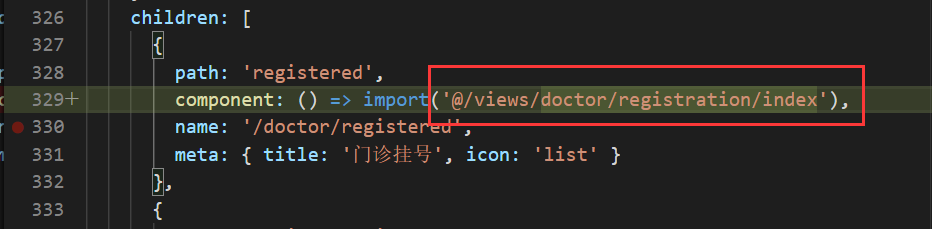


# 门诊挂号

## 创建src/api/doctor/registration.js

|  |
| --- |
| import request from '@/utils/request'  // 查询挂号列表  export function listDeptForScheduling(query) {    return request({      url: '/doctor/registration/listDeptForScheduling',      method: 'get',      params: query    })  }  // 根据身份证号查询患者信息  export function getPatientByIdCard(idCard) {    return request({      url: '/doctor/registration/getPatientByIdCard/' + idCard,      method: 'get'    })  }  // 保存挂号信息  export function addRegistration(data) {    return request({      url: '/doctor/registration/addRegistration',      method: 'post',      data: data    })  }  // 查询挂号信息  export function queryRegistrationForPage(query) {    return request({      url: '/doctor/registration/queryRegistrationForPage',      method: 'get',      params: query    })  }  // 收费  export function collectFee(registrationId) {    return request({      url: '/doctor/registration/collectFee/' + registrationId,      method: 'post'    })  } |

## 修改src/routor/index.js



## 创建src/views/doctor/registration/index.vue

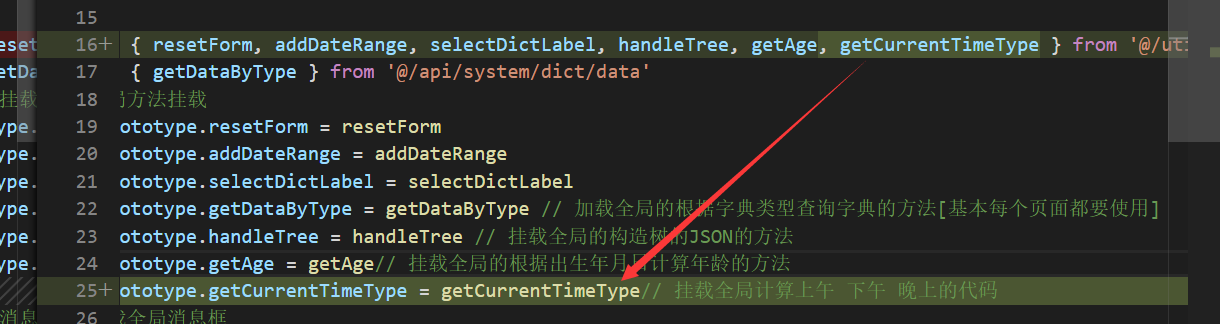
|  |
| --- |
| <template>    <div class="app-container">      <!-- 患者信息搜索开始 -->      <el-card class="box-card">        <el-form ref="queryForm" :model="patientParams" label-width="70px">          <el-row :gutter="24">            <el-col :span="12">              <el-form-item label="身份证号" porp="idCard">                <el-input                  v-model="patientParams.idCard"                  placeholder="请输入患者身份证号"                  clearable                  size="small"                />              </el-form-item>            </el-col>            <el-col :span="12">              <el-form-item>                <el-button type="primary" icon="el-icon-search" size="mini" @click="handleIdCardQuery">加载身份证号</el-button>                <el-button type="warning" icon="el-icon-search" size="mini" @click="handleIdCardQuery">加载患者信息</el-button>              </el-form-item>            </el-col>          </el-row>        </el-form>      </el-card>      <!-- 患者信息搜索结束 -->      <!-- 患者基本信息表单开始 -->      <el-card class="box-card">        <el-form ref="form" :model="patientParams" :rules="rules" :inline="true" label-width="88px">          <el-form-item label="身份证号" prop="idCard">            <el-input              v-model="patientParams.idCard"              placeholder="请输身份证号"              clearable              size="small"              style="width:200px"            />          </el-form-item>          <el-form-item label="患者姓名" prop="name">            <el-input              v-model="patientParams.name"              placeholder="请输入患者姓名"              clearable              size="small"              style="width:200px"            />          </el-form-item>          <el-form-item label="患者电话" prop="phone">            <el-input              v-model="patientParams.phone"              placeholder="请输入患者电话"              clearable              size="small"              style="width:200px"            />          </el-form-item>          <el-form-item label="性别" prop="sex">            <el-radio-group v-model="patientParams.sex">              <el-radio                v-for="d in sexOptions"                :key="d.dictValue"                :label="d.dictValue"                :value="d.dictValue"              >{{ d.dictLabel }}</el-radio>            </el-radio-group>          </el-form-item>          <el-form-item label="出生日期" prop="birthday">            <el-date-picker              v-model="patientParams.birthday"              size="small"              style="width:200px"              value-format="yyyy-MM-dd"              type="date"            />          </el-form-item>          <el-form-item label="患者年龄" prop="age">            <el-input              v-model="patientParams.age"              :readonly="true"              placeholder="请输入患者年龄"              clearable              size="small"              style="width:200px"            />          </el-form-item>          <el-form-item label="地址信息" prop="address">            <el-input              v-model="patientParams.address"              placeholder="请输入地址信息"              clearable              style="width:500px"              size="small"            />          </el-form-item>        </el-form>      </el-card>      <!-- 患者基本信息表单结束 -->      <!-- 挂号项目栏开始 -->      <el-card class="box-card">        <el-row>          <el-col :span="12">            <el-radio-group v-model="queryDeptParams.regItemId" @change="registeredItemChange">              <el-radio-button                v-for="d in regItemOptions"                :key="d.regItemId"                :label="d.regItemId"                :value="d.regItemId"              >                {{ d.regItemName }}              </el-radio-button>            </el-radio-group>          </el-col>          <el-col :span="12">            <div style="float:right">              <span style="margin-left:20px">挂号费:￥{{ queryDeptParams.regItemAmount }}</span>              <el-button type="danger" icon="el-icon-search" :disabled="single" size="small" @click="handleRegistration">挂号收费</el-button>            </div>          </el-col>        </el-row>      </el-card>      <!-- 挂号项目栏结束 -->      <el-card>        <el-form ref="queryDeptForm" :model="queryDeptParams" :inline="true" label-width="68px">          <el-form-item label="所属科室" prop="deptId">            <el-select              v-model="queryDeptParams.deptId"              placeholder="请选择所属科室"              clearable              size="small"              style="width:200px"            >              <el-option                v-for="d in deptOptions"                :key="d.deptId"                :label="d.deptName"                :value="d.deptId"              />            </el-select>          </el-form-item>          <el-form-item label="挂号类型" prop="schedulingType">            <el-select              v-model="queryDeptParams.schedulingType"              placeholder="请选择挂号类型"              clearable              size="small"              style="width:200px"            >              <el-option                v-for="dict in schedulingTypeOptions"                :key="dict.dictValue"                :label="dict.dictLabel"                :value="dict.dictValue"              />            </el-select>          </el-form-item>          <el-form-item label="挂号时段" prop="subsectionType">            <el-select              v-model="queryDeptParams.subsectionType"              placeholder="请选择挂号时段"              clearable              size="small"              style="width:200px"            >              <el-option                v-for="dict in subsectionTypeOptions"                :key="dict.dictValue"                :label="dict.dictLabel"                :value="dict.dictValue"              />            </el-select>          </el-form-item>          <el-form-item label="挂号时间" prop="schedulingDay">            <el-date-picker              v-model="queryDeptParams.schedulingDay"              size="small"              style="width:240px"              value-format="yyyy-MM-dd"              type="date"            />          </el-form-item>          <el-form-item>            <el-button type="primary" icon="el-icon-search" size="mini" @click="handleDeptQuery">搜索</el-button>            <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetDeptQuery">重置</el-button>          </el-form-item>        </el-form>        <!-- 部门数据表格开始 -->        <el-table v-loading="loading" border :data="deptTableList" highlight-current-row @selection-change="handleSelectionChange">          <el-table-column type="selection" width="55" align="center" />          <el-table-column label="科室ID" align="center" prop="deptId" />          <el-table-column label="科室名称" align="center" prop="deptName" />          <el-table-column label="当前号数" align="center" prop="regNumber" />        </el-table>        <!-- 部门数据表格结束 -->      </el-card></div>  </template>  <script>  import { getPatientByIdCard, listDeptForScheduling, addRegistration, collectFee } from '@/api/docter/registration'  import { selectAllDept } from '@/api/system/dept'  import { selectAllRegisteredItem } from '@/api/system/registeredItem'  export default {    data() {      return {        // 遮罩层        loading: false,        // 选中部门Ids        ids: [],        // 非单个禁用        single: true,        // 科室下拉列表数据        deptOptions: [],        // 性别数据字典        sexOptions: [],        // 挂号类型        schedulingTypeOptions: [],        // 排班时段        subsectionTypeOptions: [],        // 挂号项目        regItemOptions: [],        // 患者信息查询参数        patientParams: {          sex: '2'        },        // 有号部门的查询条件        queryDeptParams: {          deptId: undefined,          schedulingType: '1',          subsectionType: undefined,          schedulingDay: new Date(),          regItemId: 1,          regItemAmount: undefined        },        // 挂号费用        regItemAmount: undefined,        // 表单校验        rules: {          name: [            { required: true, message: '患者姓名不能为空', trigger: 'blur' }          ],          phone: [            { required: true, message: '手机号不能为空', trigger: 'blur' },            { pattern: /^1[3|4|5|7|8|9][0-9]\d{8}$/, message: '请输入正确的手机号', trigger: 'blur' }          ],          idCard: [            { required: true, message: '身份证号不能为空', trigger: 'blur' }          ]        },        // 部门数据        deptTableList: []      }    },    watch: {      patientParams: {        handler: function() {          if (this.patientParams.birthday !== undefined) {            this.patientParams.age = this.getAge(this.patientParams.birthday.substring(0, 10))          }        },        deep: true      }    },    created() {      // 查询性别      selectAllDept().then(res => {        this.deptOptions = res.data      })      // 加载排班类型 1 门诊  2急诊      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 排班时段 1 2 3 上午 下午 晚上      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 加载用户性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })      // 查询所有可用的挂号项目      selectAllRegisteredItem().then(res => {        this.regItemOptions = res.data        this.queryDeptParams.regItemId = this.regItemOptions[0].regItemId        this.queryDeptParams.regItemAmount = this.regItemOptions[0].regItemFee      })      // 设置查询时段      const subsectionType = this.getCurrentTimeType()      this.queryDeptParams.subsectionType = subsectionType      // 加载部门数据      this.getDeptForScheduling()    },    methods: {      // 根据输入的身份证号加载患者数据      handleIdCardQuery() {        if (this.patientParams.idCard === undefined) {          this.msgError('请输入要查询的身份证号')          return        }        this.loading = true        getPatientByIdCard(this.patientParams.idCard).then(res => {          console.log(res.data)          this.patientParams = res.data          this.patientParams.age = this.getAge(res.data.birthday.substring(0, 10))          this.loading = false        }).catch(() => {          this.loading = false          this.patientParams = { sex: '2' }        })      },      // 更新挂号项目类型事件      registeredItemChange(id) {        this.regItemOptions.filter(item => {          if (id === item.regItemId) {            this.queryDeptParams.regItemId = item.regItemId            this.queryDeptParams.regItemAmount = item.regItemFee          }        })        // 动态显示挂号信息        if (id === 1 || id === 2) {          this.queryDeptParams.schedulingType = '1'        } else {          this.queryDeptParams.schedulingType = '2'        }        // 重新查询有号的部门        this.getDeptForScheduling()      },      getDeptForScheduling() {        this.loading = true        listDeptForScheduling(this.queryDeptParams).then(res => {          this.deptTableList = res.data          this.loading = false        })      },      // 查询      handleDeptQuery() {        this.getDeptForScheduling()      },      // 重置      resetDeptQuery() {        this.resetForm('queryDeptForm')        this.getDeptForScheduling()      },      // 表格选择中行事件      handleSelectionChange(selection) {        this.ids = selection.map(item => item.deptId)        this.single = selection.length !== 1      },      // 挂号收费      handleRegistration() {        const tx = this        tx.$refs['form'].validate(vaid => {          console.log(vaid)          if (vaid) {            tx.$confirm('是否确认给【' + this.patientParams.name + '】进行挂号?', '提示', {              confirmButtonText: '确定',              cancelButtonText: '取消',              closeOnClickModal: false,              type: 'waring',              center: true            }).then(() => {              // 取出要挂号的部门              const deptId = tx.ids[0]              // 组装要提交到后台的挂号数据              const data = { 'patientDto': tx.patientParams, 'registrationDto': {                deptId: deptId,                regItemId: tx.queryDeptParams.regItemId,                regItemAmount: tx.queryDeptParams.regItemAmount,                visitDate: this.moment(tx.queryDeptParams.schedulingDay).format('YYYY-MM-DD'),                schedulingType: tx.queryDeptParams.schedulingType,                subsectionType: tx.queryDeptParams.subsectionType              }}              // 提交数据到后台              addRegistration(data).then(res => {                const regId = res.data                tx.msgSuccess('挂号成功 挂号单号为【' + regId + '】')                // 清空页面上的数据                tx.resetDeptQuery()                tx.patientParams = { sex: '2' }                tx.$confirm('是否确认给挂号ID为【' + regId + '】的挂号收费?', '提示', {                  confirmButtonText: '确定',                  cancelButtonText: '取消',                  closeOnClickModal: false,                  type: 'waring',                  center: true                }).then(() => {                  // 重新查询挂号列表                  collectFee(regId).then(r => {                    tx.msgSuccess('收费成功')                    tx.getDeptForScheduling()                  }).catch(() => {                    tx.msgError('收费失败')                  })                }).catch(() => {                  tx.msgError('收费取消')                })              }).catch(() => {                tx.msgError('挂号失败')              })            }).catch(() => {              tx.msgError('挂号已取消')            })          }        })      }    }  }  </script>  <style scoped>    .box-card{      padding: 0px;      margin: 0px;      margin-bottom: 8px;    }  </style> |

## 解决挂号时段选择问题

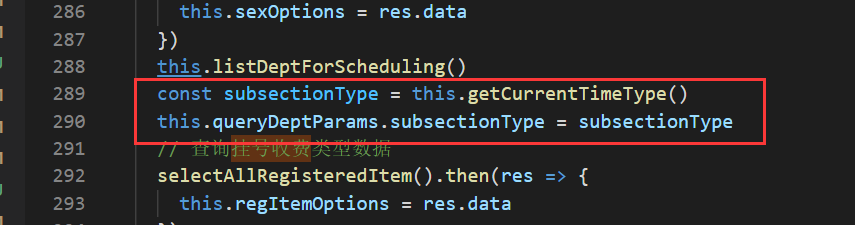
### 修改src/views/his-utils.js

|  |
| --- |
| // 判断当前时间是上午1  下午2  晚上3  // 根据出生日期计算年龄 1990-01-01  export function getCurrentTimeType() {    const now = new Date()    const hour = now.getHours()    if (hour >= 6 && hour < 12) {      return '1'    } else if (hour >= 12 && hour < 18) {      return '2'    } else if (hour >= 0 && hour < 6) {      return '3'    }  } |

### 修改src/main.js



### 创建src/views/doctor/registration/index.vue



## 格式化日期问题

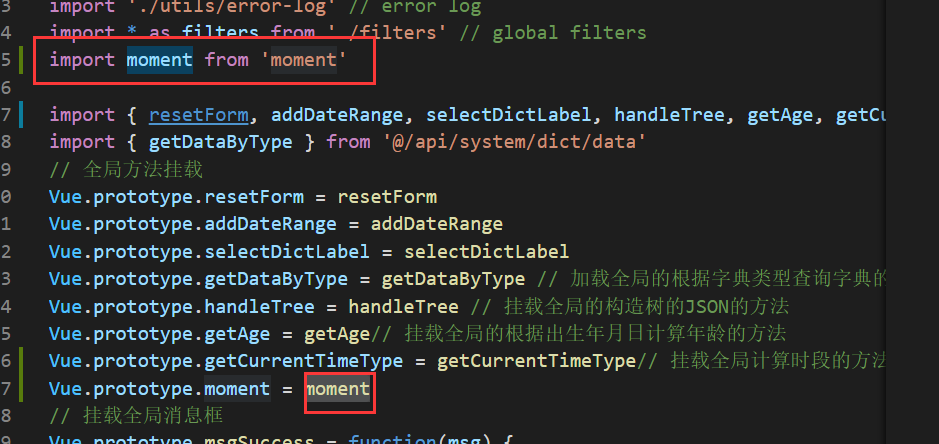
vue的方法 更简单

　　使用vue的 moment

　　安装 npm install moment --save

　　在main.js中全局引入 import moment from 'moment'

Vue.prototype.moment = moment

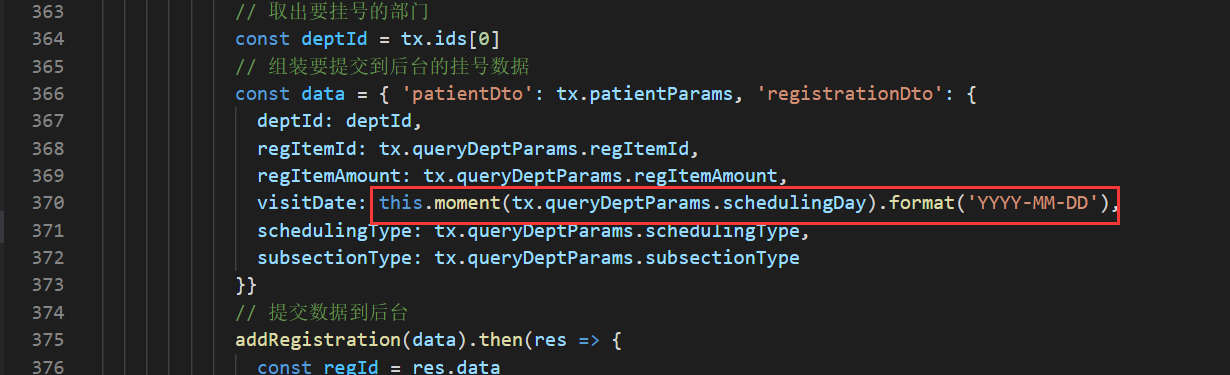


　　则在别的文件就可以直接调用了

getDate() {

console.log(this.$moment(new Date()).format('YYYY-MM-DD HH:mm:ss'))

},　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　输出结果 为 　　2020-4-20 18:04:42



# 门诊列表

## 修改src/api/doctor/registration.js

|  |
| --- |
| // 查询挂号信息  export function queryRegistrationForPage(query) {    return request({      url: '/doctor/registration/queryRegistrationForPage',      method: 'get',      params: query    })  }  // 作废  export function doInvalid(registrationId) {    return request({      url: '/doctor/registration/doInvalid/' + registrationId,      method: 'post'    })  }  // 退号  export function doReturn(registrationId) {    return request({      url: '/doctor/registration/doReturn/' + registrationId,      method: 'post'    })  } |

## 修改src/routor/index.js



## 创建src/views/doctor/registration/list.vue

|  |
| --- |
| <template>    <div class="app-container">      <!-- 查询条件开始 -->      <el-form ref="queryForm" :model="queryParams" :inline="true" label-width="68px">        <el-form-item label="所属部门" prop="deptId">          <el-select            v-model="queryParams.deptId"            placeholder="请选择所属部门"            clearable            size="small"            style="width:200px"          >            <el-option              v-for="d in deptList"              :key="d.deptId"              :label="d.deptName"              :value="d.deptId"            />          </el-select>        </el-form-item>        <el-form-item label="患者名称" prop="patientName">          <el-input            v-model="queryParams.patientName"            placeholder="请输入患者名称"            clearable            size="small"            style="width:200px"          />        </el-form-item>        <el-form-item label="挂号类型" prop="schedulingType">          <el-select            v-model="queryParams.schedulingType"            placeholder="可用状态"            clearable            size="small"            style="width:200px"          >            <el-option              v-for="dict in schedulingTypeOptions"              :key="dict.dictValue"              :label="dict.dictLabel"              :value="dict.dictValue"            />          </el-select>        </el-form-item>        <el-form-item label="挂号时段" prop="subsectionType">          <el-select            v-model="queryParams.subsectionType"            placeholder="挂号时段"            clearable            size="small"            style="width:200px"          >            <el-option              v-for="dict in subsectionTypeOptions"              :key="dict.dictValue"              :label="dict.dictLabel"              :value="dict.dictValue"            />          </el-select>        </el-form-item>        <el-form-item label="挂号状态" prop="regStatus">          <el-select            v-model="queryParams.regStatus"            placeholder="挂号状态"            clearable            size="small"            style="width:200px"          >            <el-option              v-for="dict in regStatusOptions"              :key="dict.dictValue"              :label="dict.dictLabel"              :value="dict.dictValue"            />          </el-select>        </el-form-item>        <el-form-item label="挂号时间" prop="queryDate">          <el-date-picker            v-model="queryParams.queryDate"            size="small"            style="width:200px"            value-format="yyyy-MM-dd"            type="date"          />        </el-form-item>        <el-form-item>          <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>          <el-button type="normal" icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>        </el-form-item>      </el-form>      <!-- 查询条件结束 -->      <!-- 数据表格开始 -->      <el-table v-loading="loading" border :data="regTableList">        <el-table-column type="expand">          <template slot-scope="props">            <el-form label-position="right" inline class="demo-table-expand">              <el-form-item label="挂号ID:">                <span>{{ props.row.regId }}</span>              </el-form-item>              <el-form-item label="挂号员:">                <span>{{ props.row.createBy }}</span>              </el-form-item>              <el-form-item label="创建时间:">                <span>{{ props.row.createTime }}</span>              </el-form-item>            </el-form>          </template>        </el-table-column>        <el-table-column label="患者姓名" align="center" prop="patientName" />        <el-table-column label="挂号科室" align="center" prop="deptId" :formatter="deptIdFormatter" />        <el-table-column label="接诊医生" align="center" prop="doctorName" />        <el-table-column label="挂号费用" align="center" prop="regItemAmount">          <template slot-scope="scope">            <span>{{ scope.row.regItemAmount|rounding }}</span>          </template>        </el-table-column>        <el-table-column label="流水编号" align="center" prop="regNumber" />        <el-table-column label="状态" align="center" prop="regStatus" :formatter="regStatusFormatter" />        <el-table-column label="就诊日期" align="center" prop="visitDate" />        <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />        <el-table-column label="挂号时段" align="center" prop="subsectionType" :formatter="subsectionTypeFormatter" />        <el-table-column label="操作" align="center" width="200">          <template slot-scope="scope">            <el-button v-if="scope.row.regStatus=='0'" type="success" icon="el-icon-check" size="mini" @click="handleCollect(scope.row)">收费</el-button>            <el-button v-if="scope.row.regStatus=='1'" type="danger" icon="el-icon-close" size="mini" @click="handleReturn(scope.row)">退号</el-button>            <el-button v-if="scope.row.regStatus=='0'" type="danger" icon="el-icon-check" size="mini" @click="handleInvalid(scope.row)">作废</el-button>          </template>        </el-table-column>      </el-table>      <!-- 数据表格结束 -->      <!-- 分页控件开始 -->      <el-pagination        v-show="total>0"        :current-page="queryParams.pageNum"        :page-sizes="[5,10,20,30]"        :page-size="queryParams.pageSize"        layout="total,sizes,prev,pager,next,jumper"        :total="total"        @size-change="handleSizeChange"        @current-change="handleCurrentChange"      />      <!-- 分页控件结束 -->    </div>  </template>  <script>  // 引入接口  import { selectAllDept } from '@/api/system/dept'  import { selectNeedSchedulingUsers } from '@/api/system/user'  import { queryRegistrationForPage, collectFee, doInvalid, doReturn } from '@/api/docter/registration'  export default {    // 过滤器    filters: {      // 保留两位小数      rounding(value) {        return value.toFixed(2)      }    },    data() {      return {        // 遮罩层        loading: false,        // 总条数        total: 0,        // 挂号数据        regTableList: [],        // 部门数据        deptList: [],        // 用户数据        userList: [],        // 挂号单状态数据字典        regStatusOptions: [],        // 门诊急诊        schedulingTypeOptions: [],        // 上午  中午  晚上        subsectionTypeOptions: [],        // 查询参数        queryParams: {          pageNum: 1,          pageSize: 10,          patientName: undefined,          deptId: undefined,          schedulingType: undefined,          subsectionType: undefined,          visitDate: undefined,          queryDate: new Date()        }      }    },    created() {      // 加载查询条件排班类型1 门诊 2 急诊 字典表数据翻译      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 加载排班时段1上午  2下午 3晚上 字典表数据翻译      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 加载挂号状态0未收费,1待就诊，2,就诊中，3，就诊完成，4，已退号，5 作废      this.getDataByType('his\_registration\_status').then(res => {        this.regStatusOptions = res.data      })      selectAllDept().then(res => {        this.deptList = res.data      })      selectNeedSchedulingUsers().then(res => {        this.userList = res.data      })      // 查询挂号信息      this.listRegistration()    },    methods: {      listRegistration() {        this.loading = true        this.queryParams.visitDate = this.moment(this.queryParams.queryDate).format('YYYY-MM-DD')        queryRegistrationForPage(this.queryParams).then(res => {          this.regTableList = res.data          this.total = res.total          this.loading = false        }).catch(() => {          this.loading = false        })      },      // 每页显示多少条的数据变化      handleSizeChange(val) {        this.queryParams.pageSize = val        this.listRegistration()      },      // 分页跳转      handleCurrentChange(val) {        this.queryParams.pageNum = val        this.listRegistration()      },      // 搜索      handleQuery() {        this.listRegistration()      },      resetQuery() {        this.resetForm('queryForm')        this.listRegistration()      },      // 翻译部门      deptIdFormatter(row, column) {        let deptName        this.deptList.filter(dept => {          if (dept.deptId === row.deptId) {            deptName = dept.deptName          }        })        return deptName      },      // 翻译排班类型      schedulingTypeFormatter(row, column) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType)      },      // 翻译排班时段      subsectionTypeFormatter(row, column) {        return this.selectDictLabel(this.subsectionTypeOptions, row.subsectionType)      },      // 翻译挂号单类型      regStatusFormatter(row, column) {        return this.selectDictLabel(this.regStatusOptions, row.regStatus)      },      // 收费      handleCollect(row) {        this.loading = true        collectFee(row.regId).then(res => {          this.loading = false          this.msgSuccess('收费成功')          this.listRegistration()        }).catch(() => {          this.msgError('收费失败')        })      },      // 作废      handleInvalid(row) {        this.loading = true        doInvalid(row.regId).then(res => {          this.loading = false          this.msgSuccess('作废成功')          this.listRegistration()        }).catch(() => {          this.msgError('作废失败')        })      },      // 退号      handleReturn(row) {        this.loading = true        doReturn(row.regId).then(res => {          this.loading = false          this.msgSuccess('退号成功')          this.listRegistration()        }).catch(() => {          this.msgError('退号失败')        })      }    }  }  </script>  <style scoped>   .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }  </style> |

# 新开就诊第一部分

## 创建src/api/doctor/care.js

|  |
| --- |
| import request from '@/utils/request'  // 加载挂号列表[待就诊]  export function queryToBeSeenRegistration(scheudlingType) {    return request({      url: '/doctor/care/queryToBeSeenRegistration/' + scheudlingType,      method: 'get'    })  }  // 加载挂号列表[就诊中]  export function queryVisitingRegistration(scheudlingType) {    return request({      url: '/doctor/care/queryVisitingRegistration/' + scheudlingType,      method: 'get'    })  }  // 加载挂号列表[就诊完成]  export function queryVisitCompletedRegistration(scheudlingType) {    return request({      url: '/doctor/care/queryVisitCompletedRegistration/' + scheudlingType,      method: 'get'    })  }  // 根据患者ID查询患者信息 患者档案信息  历史病例  export function getPatientAllMessageByPatientId(patientId) {    return request({      url: '/doctor/care/getPatientAllMessageByPatientId/' + patientId,      method: 'get'    })  }  // 接诊  export function receivePatient(regId) {    return request({      url: '/doctor/care/receivePatient/' + regId,      method: 'post'    })  } |

## 修改src/routor/index.js



## 创建src/views/doctor/care/index.vue

|  |
| --- |
| <template>    <div v-loading="loading" class="app-container">      <!-- 选择门诊和急诊的选项开始 -->      <el-card style="margin-bottom:5px;text-align:center">        <el-radio-group v-model="schedulingType" @change="schedulingTypeChange">          <el-radio-button            v-for="dict in schedulingTypeOptions"            :key="dict.dictValue"            :label="dict.dictValue"            :value="dict.dictValue"          >            {{ dict.dictLabel }}          </el-radio-button>        </el-radio-group>      </el-card>      <!-- 选择门诊和急诊的选项结束 -->      <!-- 下面的整体卡片开始 -->      <el-row :gutter="5">        <el-col :span="7">          <!-- 左边的患者信息开始 -->          <el-card>            <el-form ref="form" :model="patientAllObj.patientObj" :inline="true" label-width="88px">              <el-form-item label="患者姓名" prop="name">                <el-input                  v-model="patientAllObj.patientObj.name"                  placeholder="请输入患者姓名"                  :disabled="true"                  style="width:100%"                  size="small"                >                  <el-button slot="append" icon="el-icon-user-solid" @click="viewRegistration" />                </el-input>              </el-form-item>              <el-form-item label="身份证号" prop="idCard">                <el-input                  v-model="patientAllObj.patientObj.idCard"                  placeholder="请输入身份证号"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者电话" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.phone"                  placeholder="请输入患者电话"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="性别" prop="sex">                <el-radio-group v-model="patientAllObj.patientObj.sex">                  <el-radio                    v-for="d in sexOptions"                    :key="d.dictValue"                    :label="d.dictValue"                    :value="d.dictValue"                  >{{ d.dictLabel }}</el-radio>                </el-radio-group>              </el-form-item>              <el-form-item label="出生日期" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.birthday"                  placeholder="请选择出生日期"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者年龄" prop="age">                <el-input                  v-model="patientAllObj.patientObj.age"                  placeholder="请输入患者年龄"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者地址" prop="address">                <el-input                  v-model="patientAllObj.patientObj.address"                  placeholder="请输入患者地址"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="过敏信息" prop="allergyInfo">                <el-input                  v-model="patientAllObj.patientObj.allergyInfo"                  placeholder="请输入过敏信息"                  type="textarea"                  :autosize="{minRows:2,maxRows:4}"                  :disabled="true"                  maxlength="100"                  show-word-limit                  style="width:100%"                  size="small"                />              </el-form-item>            </el-form>            <el-tabs :stretch="true" type="card">              <el-tab-pane label="患者档案">                <div v-if="patientAllObj.patientObj.isFinal==='1'">                  <div class="item">                    紧急联系人:<span v-text="patientAllObj.patientFileObj.emergencyContactName" />                  </div>                  <div class="item">                    紧急联系人电话:<span v-text="patientAllObj.patientFileObj.emergencyContactPhone" />                  </div>                  <div class="item">                    关系: <span v-text="patientAllObj.patientFileObj.emergencyContactRelation" />                  </div>                  <div class="item">                    左耳听力:<span v-text="patientAllObj.patientFileObj.leftEarHearing" />                  </div>                  <div class="item">                    右耳听力:<span v-text="patientAllObj.patientFileObj.rightEarHearing" />                  </div>                  <div class="item">                    左眼视力:<span v-text="patientAllObj.patientFileObj.leftVision" />                  </div>                  <div class="item">                    右眼视力:<span v-text="patientAllObj.patientFileObj.rightVision" />                  </div>                  <div class="item">                    身高:<span v-text="patientAllObj.patientFileObj.height" />                  </div>                  <div class="item">                    体重:<span v-text="patientAllObj.patientFileObj.weight" />                  </div>                  <div class="item">                    血型:<span v-text="patientAllObj.patientFileObj.bloodType" />                  </div>                  <div class="item">                    个人史:<span v-text="patientAllObj.patientFileObj.personalInfo" />                  </div>                  <div class="item">                    家族史:<span v-text="patientAllObj.patientFileObj.familyInfo" />                  </div>                  <div class="item">                    档案创建时间:<span v-text="patientAllObj.patientFileObj.createTime" />                  </div>                  <div class="item">                    档案更新时间:<span v-text="patientAllObj.patientFileObj.updateTime" />                  </div>                </div>                <div v-else style="text-align:center">                  暂无患者档案信息                </div>              </el-tab-pane>              <el-tab-pane label="历史病历">                <div v-if="patientAllObj.careHistoryObjList.length>0">                  <el-collapse accordion>                    <el-collapse-item v-for="(item,index) in patientAllObj.careHistoryObjList" :key="index">                      <template slot="title">                        就诊部门:【{{ item.deptName }}】 就诊时间:{{ item.careDate }}                      </template>                      <div class="item">                        主诉:{{ item.caseTitle }}                      </div>                      <div class="item">                        发病日期:{{ item.caseDate }}                      </div>                      <div class="item">                        诊断信息:{{ item.caseResult }}                      </div>                      <div class="item">                        医生建议:{{ item.doctorTips }}                      </div>                      <div class="item">                        备注:{{ item.remark }}                      </div>                      <div class="item">                        接诊类型:{{ item.receiveType==='0'?'初诊':'复诊' }}                      </div>                      <div class="item">                        是否传染:{{ item.isContagious==='0'?'否':'是' }}                      </div>                    </el-collapse-item>                  </el-collapse>                </div>                <div v-else style="text-align:center">                  暂无患者病历信息                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 左边的患者信息结束 -->        </el-col>        <el-col :span="17">          <!-- 右边的病历，处方开始 -->          <el-card>            <el-row :gutter="5">              <el-col :span="12">                挂号单号:                <span v-if="careHistory.regId===undefined" style="color:red">选择患者后显示</span>                <span v-else style="color:blue">{{ careHistory.regId }}</span>                <br>                病历编号:                <span style="color:red">保存病例后显示</span>              </el-col>              <el-col :span="12" style="text-align:right">                <el-button type="primary" :disabled="careHistory.regId===undefined" icon="el-icon-check">保存病历</el-button>                <el-button type="danger" :disabled="careHistory.regId===undefined" icon="el-icon-finished">完成就诊</el-button>              </el-col>            </el-row>          </el-card>          <!-- 右边的病历，处方结束 -->        </el-col>      </el-row>      <!-- 下面的整体卡片结束 -->      <!-- 弹出选择患者的层开始 -->      <el-dialog        title="请选择挂号患者"        :visible.sync="openRegistration"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <el-tabs v-model="activeName" :stretch="true" type="card" @tab-click="handleRegistrationTabClick">          <el-tab-pane label="待就诊列表" name="t1">            <el-table v-loading="tableLoading" border :data="toBeSeenRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleVisit(scope.row)">接诊</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊中列表" name="t2">            <el-table v-loading="tableLoading" border :data="visitingRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊完成列表" name="t3">            <el-table v-loading="tableLoading" border :data="visitCompletedRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>        </el-tabs>      </el-dialog>      <!-- 弹出选择患者的层结束 -->    </div>  </template>  <script>  import { queryToBeSeenRegistration, queryVisitingRegistration, queryVisitCompletedRegistration, receivePatient, getPatientAllMessageByPatientId } from '@/api/docter/care'  export default {    data() {      return {        // 总的遮罩层        loading: false,        // 表格的遮罩层        tableLoading: false,        // 门诊急诊的默认值        schedulingType: '1',        // 就诊类型的字典数据        schedulingTypeOptions: [],        // 排班类型的字典数据        subsectionTypeOptions: [],        // 性别数据字典        sexOptions: [],        // 是否打开选择挂号患者的弹层        openRegistration: false,        // 当前选中的挂号选项卡的        activeName: 't1',        // 待就诊数据        toBeSeenRegistration: [],        // 就诊中的数据        visitingRegistration: [],        // 就诊完成的数据        visitCompletedRegistration: [],        // 左边患者数据        patientAllObj: {          patientObj: {},          patientFileObj: {},          careHistoryObjList: []        },        // 提交到数据库的对象        careHistory: {        }      }    },    created() {      // 加载排班类型 1 门诊  2急诊      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 排班时段 1 2 3 上午 下午 晚上      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 加载性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })    },    methods: {      // 门诊急诊切换事件      schedulingTypeChange(value) {        // console.log(value)        this.schedulingType = value      },      // 打开选择挂号患者的弹出层      viewRegistration() {        this.activeName = 't1'        this.openRegistration = true        this.queryToBeSeenRegistration()      },      // 挂号患者弹出层的选项卡change事件      handleRegistrationTabClick(tab, event) {        if (tab.name === 't1') {          this.queryToBeSeenRegistration()        } else if (tab.name === 't2') {          this.queryVisitingRegistration()        } else if (tab.name === 't3') {          this.queryVisitCompletedRegistration()        }      },      // 查询待就诊的数据      queryToBeSeenRegistration() {        this.tableLoading = true        queryToBeSeenRegistration(this.schedulingType).then(res => {          this.toBeSeenRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊中的数据      queryVisitingRegistration() {        this.tableLoading = true        queryVisitingRegistration(this.schedulingType).then(res => {          this.visitingRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊完成的数据      queryVisitCompletedRegistration() {        this.tableLoading = true        queryVisitCompletedRegistration(this.schedulingType).then(res => {          this.visitCompletedRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 翻译挂号类型      schedulingTypeFormatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType)      },      // 接诊      handleVisit(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 接诊，更新挂号的状态为就诊中并写入接诊的医生信息        this.loading = true        receivePatient(this.careHistory.regId).then(res => {          this.loading = false          this.msgSuccess('接诊成功')          // 根据患者ID查询患者信息档案信息 病历信息          getPatientAllMessageByPatientId(patientId).then(res => {            this.patientAllObj.patientObj = res.data.patient            this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)            this.patientAllObj.patientFileObj = res.data.patientFile            this.patientAllObj.careHistoryObjList = res.data.careHistoryList            this.openRegistration = false            this.loading = false          }).catch(() => {            this.msgError('查询患者信息失败')            this.loading = false          })        }).catch(() => {          this.msgError('接诊失败')          this.loading = false        })      },      // 载入      handleLoading(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 根据患者ID查询患者信息档案信息 病历信息        getPatientAllMessageByPatientId(patientId).then(res => {          this.patientAllObj.patientObj = res.data.patient          this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)          this.patientAllObj.patientFileObj = res.data.patientFile          this.patientAllObj.careHistoryObjList = res.data.careHistoryList          this.openRegistration = false          this.loading = false          // 查询当前挂号单之前对应的病例信息        }).catch(() => {          this.msgError('载入失败')          this.loading = false        })      }    }  }  </script>  <style scoped>    .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }    .item{      font-size: 14px;      margin-bottom: 8px;    }  </style> |

# 新开就诊第二部分

## 保存病历

### 修改src/api/doctor/care.js

|  |
| --- |
| // 保存病例信息  export function saveCareHistory(data) {    return request({      url: '/doctor/care/saveCareHistory',      method: 'post',      data: data    })  } |

### 修改src/views/doctor/care/index.vue

|  |
| --- |
| <template>    <div v-loading="loading" class="app-container">      <!-- 选择门诊和急诊的选项开始 -->      <el-card style="margin-bottom:5px;text-align:center">        <el-radio-group v-model="schedulingType" @change="schedulingTypeChange">          <el-radio-button            v-for="dict in schedulingTypeOptions"            :key="dict.dictValue"            :label="dict.dictValue"            :value="dict.dictValue"          >            {{ dict.dictLabel }}          </el-radio-button>        </el-radio-group>      </el-card>      <!-- 选择门诊和急诊的选项结束 -->      <!-- 下面的整体卡片开始 -->      <el-row :gutter="5">        <el-col :span="7">          <!-- 左边的患者信息开始 -->          <el-card>            <el-form ref="form" :model="patientAllObj.patientObj" :inline="true" label-width="88px">              <el-form-item label="患者姓名" prop="name">                <el-input                  v-model="patientAllObj.patientObj.name"                  placeholder="请输入患者姓名"                  :disabled="true"                  style="width:100%"                  size="small"                >                  <el-button slot="append" icon="el-icon-user-solid" @click="viewRegistration" />                </el-input>              </el-form-item>              <el-form-item label="身份证号" prop="idCard">                <el-input                  v-model="patientAllObj.patientObj.idCard"                  placeholder="请输入身份证号"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者电话" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.phone"                  placeholder="请输入患者电话"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="性别" prop="sex">                <el-radio-group v-model="patientAllObj.patientObj.sex">                  <el-radio                    v-for="d in sexOptions"                    :key="d.dictValue"                    :label="d.dictValue"                    :value="d.dictValue"                  >{{ d.dictLabel }}</el-radio>                </el-radio-group>              </el-form-item>              <el-form-item label="出生日期" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.birthday"                  placeholder="请选择出生日期"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者年龄" prop="age">                <el-input                  v-model="patientAllObj.patientObj.age"                  placeholder="请输入患者年龄"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者地址" prop="address">                <el-input                  v-model="patientAllObj.patientObj.address"                  placeholder="请输入患者地址"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="过敏信息" prop="allergyInfo">                <el-input                  v-model="patientAllObj.patientObj.allergyInfo"                  placeholder="请输入过敏信息"                  type="textarea"                  :autosize="{minRows:2,maxRows:4}"                  :disabled="true"                  maxlength="100"                  show-word-limit                  style="width:100%"                  size="small"                />              </el-form-item>            </el-form>            <el-tabs :stretch="true" type="card">              <el-tab-pane label="患者档案">                <div v-if="patientAllObj.patientObj.isFinal==='1'">                  <div class="item">                    紧急联系人:<span v-text="patientAllObj.patientFileObj.emergencyContactName" />                  </div>                  <div class="item">                    紧急联系人电话:<span v-text="patientAllObj.patientFileObj.emergencyContactPhone" />                  </div>                  <div class="item">                    关系: <span v-text="patientAllObj.patientFileObj.emergencyContactRelation" />                  </div>                  <div class="item">                    左耳听力:<span v-text="patientAllObj.patientFileObj.leftEarHearing" />                  </div>                  <div class="item">                    右耳听力:<span v-text="patientAllObj.patientFileObj.rightEarHearing" />                  </div>                  <div class="item">                    左眼视力:<span v-text="patientAllObj.patientFileObj.leftVision" />                  </div>                  <div class="item">                    右眼视力:<span v-text="patientAllObj.patientFileObj.rightVision" />                  </div>                  <div class="item">                    身高:<span v-text="patientAllObj.patientFileObj.height" />                  </div>                  <div class="item">                    体重:<span v-text="patientAllObj.patientFileObj.weight" />                  </div>                  <div class="item">                    血型:<span v-text="patientAllObj.patientFileObj.bloodType" />                  </div>                  <div class="item">                    个人史:<span v-text="patientAllObj.patientFileObj.personalInfo" />                  </div>                  <div class="item">                    家族史:<span v-text="patientAllObj.patientFileObj.familyInfo" />                  </div>                  <div class="item">                    档案创建时间:<span v-text="patientAllObj.patientFileObj.createTime" />                  </div>                  <div class="item">                    档案更新时间:<span v-text="patientAllObj.patientFileObj.updateTime" />                  </div>                </div>                <div v-else style="text-align:center">                  暂无患者档案信息                </div>              </el-tab-pane>              <el-tab-pane label="历史病历">                <div v-if="patientAllObj.careHistoryObjList.length>0">                  <el-collapse accordion>                    <el-collapse-item v-for="(item,index) in patientAllObj.careHistoryObjList" :key="index">                      <template slot="title">                        就诊部门:【{{ item.deptName }}】 就诊时间:{{ item.careDate }}                      </template>                      <div class="item">                        主诉:{{ item.caseTitle }}                      </div>                      <div class="item">                        发病日期:{{ item.caseDate }}                      </div>                      <div class="item">                        诊断信息:{{ item.caseResult }}                      </div>                      <div class="item">                        医生建议:{{ item.doctorTips }}                      </div>                      <div class="item">                        备注:{{ item.remark }}                      </div>                      <div class="item">                        接诊类型:{{ item.receiveType==='0'?'初诊':'复诊' }}                      </div>                      <div class="item">                        是否传染:{{ item.isContagious==='0'?'否':'是' }}                      </div>                    </el-collapse-item>                  </el-collapse>                </div>                <div v-else style="text-align:center">                  暂无患者病历信息                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 左边的患者信息结束 -->        </el-col>        <el-col :span="17">          <!-- 右边的病历，处方开始 -->          <el-card>            <el-row :gutter="5">              <el-col :span="12">                挂号单号:                <span v-if="careHistory.regId===undefined" style="color:red">选择患者后显示</span>                <span v-else style="color:blue">{{ careHistory.regId }}</span>                <br>                病历编号:                <span v-if="careHistory.chId===undefined" style="color:red">保存病例后显示</span>                <span v-else style="color:blue">{{ careHistory.chId }}</span>              </el-col>              <el-col :span="12" style="text-align:right">                <el-button type="primary" :disabled="careHistory.regId===undefined" icon="el-icon-check" @click="handelSaveCareHistory">保存病历</el-button>                <el-button type="danger" :disabled="careHistory.regId===undefined" icon="el-icon-finished">完成就诊</el-button>              </el-col>            </el-row>          </el-card>          <el-card style="margin-top:3px">            <el-tabs v-model="careActiveName" type="card">              <el-tab-pane label="病历" name="c1">                <el-card>                  <el-form ref="form" :model="careHistory" :inline="true" label-width="88px">                    <el-form-item label="发病日期" prop="caseDateObj">                      <el-date-picker                        v-model="careHistory.caseDateObj"                        value-format="yyyy-MM-dd"                        size="small"                      />                    </el-form-item>                    <el-form-item label="接诊类型" prop="receiveType">                      <el-select                        v-model="careHistory.receiveType"                        placeholder="接诊类型"                        size="small"                      >                        <el-option                          v-for="dict in receiveTypeOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                    <el-form-item label="是否传染" prop="isContagious">                      <el-select                        v-model="careHistory.isContagious"                        placeholder="是否传染"                        size="small"                      >                        <el-option                          v-for="dict in isContagiousOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                  </el-form>                  <el-form ref="form" :model="careHistory" label-width="88px">                    <el-form-item label="主诉" :inline="false" prop="caseTitle">                      <el-input                        v-model="careHistory.caseTitle"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入主诉"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="诊断信息" :inline="false" prop="caseResult">                      <el-input                        v-model="careHistory.caseResult"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入诊断信息"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="医生建议" :inline="false" prop="doctorTips">                      <el-input                        v-model="careHistory.doctorTips"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入医生建议"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="备注" :inline="false" prop="remark">                      <el-input                        v-model="careHistory.remark"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入备注"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                  </el-form>                </el-card>              </el-tab-pane>              <el-tab-pane label="处方" name="c2">                处方              </el-tab-pane>            </el-tabs></el-card>          <!-- 右边的病历，处方结束 -->        </el-col>      </el-row>      <!-- 下面的整体卡片结束 -->      <!-- 弹出选择患者的层开始 -->      <el-dialog        title="请选择挂号患者"        :visible.sync="openRegistration"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <el-tabs v-model="activeName" :stretch="true" type="card" @tab-click="handleRegistrationTabClick">          <el-tab-pane label="待就诊列表" name="t1">            <el-table v-loading="tableLoading" border :data="toBeSeenRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleVisit(scope.row)">接诊</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊中列表" name="t2">            <el-table v-loading="tableLoading" border :data="visitingRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊完成列表" name="t3">            <el-table v-loading="tableLoading" border :data="visitCompletedRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>        </el-tabs>      </el-dialog>      <!-- 弹出选择患者的层结束 -->    </div>  </template>  <script>  import { queryToBeSeenRegistration, queryVisitingRegistration, queryVisitCompletedRegistration, receivePatient,    getPatientAllMessageByPatientId, saveCareHistory  } from '@/api/docter/care'  export default {    data() {      return {        // 总的遮罩层        loading: false,        // 表格的遮罩层        tableLoading: false,        // 门诊急诊的默认值        schedulingType: '1',        // 就诊类型的字典数据        schedulingTypeOptions: [],        // 排班类型的字典数据        subsectionTypeOptions: [],        // 性别数据字典        sexOptions: [],        // 接诊类型字典        receiveTypeOptions: [],        // 是否传染字典        isContagiousOptions: [],        // 是否打开选择挂号患者的弹层        openRegistration: false,        // 当前选中的挂号选项卡的        activeName: 't1',        // 右边的选项卡        careActiveName: 'c1',        // 待就诊数据        toBeSeenRegistration: [],        // 就诊中的数据        visitingRegistration: [],        // 就诊完成的数据        visitCompletedRegistration: [],        // 左边患者数据        patientAllObj: {          patientObj: {},          patientFileObj: {},          careHistoryObjList: []        },        // 提交到数据库的对象        careHistory: {          // 当前就诊中的挂号单ID          regId: undefined,          chId: undefined,          caseDate: undefined,          caseDateObj: new Date(),          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        }      }    },    created() {      // 加载排班类型 1 门诊  2急诊      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 排班时段 1 2 3 上午 下午 晚上      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 加载性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })      //  加载接诊类型字典      this.getDataByType('his\_receive\_type').then(res => {        this.receiveTypeOptions = res.data      })      //  加载是否传染字典      this.getDataByType('his\_contagious\_status').then(res => {        this.isContagiousOptions = res.data      })    },    methods: {      // 门诊急诊切换事件      schedulingTypeChange(value) {        // console.log(value)        this.schedulingType = value      },      // 打开选择挂号患者的弹出层      viewRegistration() {        this.activeName = 't1'        this.openRegistration = true        this.queryToBeSeenRegistration()      },      // 挂号患者弹出层的选项卡change事件      handleRegistrationTabClick(tab, event) {        if (tab.name === 't1') {          this.queryToBeSeenRegistration()        } else if (tab.name === 't2') {          this.queryVisitingRegistration()        } else if (tab.name === 't3') {          this.queryVisitCompletedRegistration()        }      },      // 查询待就诊的数据      queryToBeSeenRegistration() {        this.tableLoading = true        queryToBeSeenRegistration(this.schedulingType).then(res => {          this.toBeSeenRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊中的数据      queryVisitingRegistration() {        this.tableLoading = true        queryVisitingRegistration(this.schedulingType).then(res => {          this.visitingRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊完成的数据      queryVisitCompletedRegistration() {        this.tableLoading = true        queryVisitCompletedRegistration(this.schedulingType).then(res => {          this.visitCompletedRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 翻译挂号类型      schedulingTypeFormatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType)      },      // 接诊      handleVisit(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 接诊，更新挂号的状态为就诊中并写入接诊的医生信息        this.loading = true        receivePatient(this.careHistory.regId).then(res => {          this.loading = false          this.msgSuccess('接诊成功')          // 根据患者ID查询患者信息档案信息 病历信息          getPatientAllMessageByPatientId(patientId).then(res => {            this.patientAllObj.patientObj = res.data.patient            this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)            this.patientAllObj.patientFileObj = res.data.patientFile            this.patientAllObj.careHistoryObjList = res.data.careHistoryList            this.openRegistration = false            this.loading = false          }).catch(() => {            this.msgError('查询患者信息失败')            this.loading = false          })        }).catch(() => {          this.msgError('接诊失败')          this.loading = false        })      },      // 载入      handleLoading(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 根据患者ID查询患者信息档案信息 病历信息        getPatientAllMessageByPatientId(patientId).then(res => {          this.patientAllObj.patientObj = res.data.patient          this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)          this.patientAllObj.patientFileObj = res.data.patientFile          this.patientAllObj.careHistoryObjList = res.data.careHistoryList          this.openRegistration = false          this.loading = false          // 查询当前挂号单之前对应的病例信息        }).catch(() => {          this.msgError('载入失败')          this.loading = false        })      },      // 保存病例      handelSaveCareHistory() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        this.careHistory.caseDate = this.moment(this.careHistory.caseDateObj).format('YYYY-MM-DD')        this.careHistory.patientId = this.patientAllObj.patientObj.patientId        this.careHistory.patientName = this.patientAllObj.patientObj.name        this.loading = true        saveCareHistory(this.careHistory).then(res => {          this.loading = false          this.msgSuccess('保存病例成功')          this.careHistory.chId = res.data.chId        }).catch(() => {          this.msgError('保存病例失败')          this.loading = false        })      }    }  }  </script>  <style scoped>    .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }    .item{      font-size: 14px;      margin-bottom: 8px;    }  </style> |

## 查询病历和处方信息

### 修改src/api/doctor/care.js

|  |
| --- |
| // 根据挂号ID查询病历信息  export function getCareHistoryByRegId(regId) {    return request({      url: '/doctor/care/getCareHistoryByRegId/' + regId,      method: 'get'    })  }  // 根据病例ID查询处方列表及详情  export function queryCareOrdersByChId(chId) {    return request({      url: '/doctor/care/queryCareOrdersByChId/' + chId,      method: 'get'    })  } |

### 修改src/views/doctor/care/index.vue

|  |
| --- |
| <template>    <div v-loading="loading" class="app-container">      <!-- 选择门诊和急诊的选项开始 -->      <el-card style="margin-bottom:5px;text-align:center">        <el-radio-group v-model="schedulingType" @change="schedulingTypeChange">          <el-radio-button            v-for="dict in schedulingTypeOptions"            :key="dict.dictValue"            :label="dict.dictValue"            :value="dict.dictValue"          >            {{ dict.dictLabel }}          </el-radio-button>        </el-radio-group>      </el-card>      <!-- 选择门诊和急诊的选项结束 -->      <!-- 下面的整体卡片开始 -->      <el-row :gutter="5">        <el-col :span="7">          <!-- 左边的患者信息开始 -->          <el-card>            <el-form ref="form" :model="patientAllObj.patientObj" :inline="true" label-width="88px">              <el-form-item label="患者姓名" prop="name">                <el-input                  v-model="patientAllObj.patientObj.name"                  placeholder="请输入患者姓名"                  :disabled="true"                  style="width:100%"                  size="small"                >                  <el-button slot="append" icon="el-icon-user-solid" @click="viewRegistration" />                </el-input>              </el-form-item>              <el-form-item label="身份证号" prop="idCard">                <el-input                  v-model="patientAllObj.patientObj.idCard"                  placeholder="请输入身份证号"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者电话" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.phone"                  placeholder="请输入患者电话"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="性别" prop="sex">                <el-radio-group v-model="patientAllObj.patientObj.sex">                  <el-radio                    v-for="d in sexOptions"                    :key="d.dictValue"                    :label="d.dictValue"                    :value="d.dictValue"                  >{{ d.dictLabel }}</el-radio>                </el-radio-group>              </el-form-item>              <el-form-item label="出生日期" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.birthday"                  placeholder="请选择出生日期"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者年龄" prop="age">                <el-input                  v-model="patientAllObj.patientObj.age"                  placeholder="请输入患者年龄"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者地址" prop="address">                <el-input                  v-model="patientAllObj.patientObj.address"                  placeholder="请输入患者地址"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="过敏信息" prop="allergyInfo">                <el-input                  v-model="patientAllObj.patientObj.allergyInfo"                  placeholder="请输入过敏信息"                  type="textarea"                  :autosize="{minRows:2,maxRows:4}"                  :disabled="true"                  maxlength="100"                  show-word-limit                  style="width:100%"                  size="small"                />              </el-form-item>            </el-form>            <el-tabs :stretch="true" type="card">              <el-tab-pane label="患者档案">                <div v-if="patientAllObj.patientObj.isFinal==='1'">                  <div class="item">                    紧急联系人:<span v-text="patientAllObj.patientFileObj.emergencyContactName" />                  </div>                  <div class="item">                    紧急联系人电话:<span v-text="patientAllObj.patientFileObj.emergencyContactPhone" />                  </div>                  <div class="item">                    关系: <span v-text="patientAllObj.patientFileObj.emergencyContactRelation" />                  </div>                  <div class="item">                    左耳听力:<span v-text="patientAllObj.patientFileObj.leftEarHearing" />                  </div>                  <div class="item">                    右耳听力:<span v-text="patientAllObj.patientFileObj.rightEarHearing" />                  </div>                  <div class="item">                    左眼视力:<span v-text="patientAllObj.patientFileObj.leftVision" />                  </div>                  <div class="item">                    右眼视力:<span v-text="patientAllObj.patientFileObj.rightVision" />                  </div>                  <div class="item">                    身高:<span v-text="patientAllObj.patientFileObj.height" />                  </div>                  <div class="item">                    体重:<span v-text="patientAllObj.patientFileObj.weight" />                  </div>                  <div class="item">                    血型:<span v-text="patientAllObj.patientFileObj.bloodType" />                  </div>                  <div class="item">                    个人史:<span v-text="patientAllObj.patientFileObj.personalInfo" />                  </div>                  <div class="item">                    家族史:<span v-text="patientAllObj.patientFileObj.familyInfo" />                  </div>                  <div class="item">                    档案创建时间:<span v-text="patientAllObj.patientFileObj.createTime" />                  </div>                  <div class="item">                    档案更新时间:<span v-text="patientAllObj.patientFileObj.updateTime" />                  </div>                </div>                <div v-else style="text-align:center">                  暂无患者档案信息                </div>              </el-tab-pane>              <el-tab-pane label="历史病历">                <div v-if="patientAllObj.careHistoryObjList.length>0">                  <el-collapse accordion>                    <el-collapse-item v-for="(item,index) in patientAllObj.careHistoryObjList" :key="index">                      <template slot="title">                        就诊部门:【{{ item.deptName }}】 就诊时间:{{ item.careDate }}                      </template>                      <div class="item">                        主诉:{{ item.caseTitle }}                      </div>                      <div class="item">                        发病日期:{{ item.caseDate }}                      </div>                      <div class="item">                        诊断信息:{{ item.caseResult }}                      </div>                      <div class="item">                        医生建议:{{ item.doctorTips }}                      </div>                      <div class="item">                        备注:{{ item.remark }}                      </div>                      <div class="item">                        接诊类型:{{ item.receiveType==='0'?'初诊':'复诊' }}                      </div>                      <div class="item">                        是否传染:{{ item.isContagious==='0'?'否':'是' }}                      </div>                    </el-collapse-item>                  </el-collapse>                </div>                <div v-else style="text-align:center">                  暂无患者病历信息                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 左边的患者信息结束 -->        </el-col>        <el-col :span="17">          <!-- 右边的病历，处方开始 -->          <el-card>            <el-row :gutter="5">              <el-col :span="12">                挂号单号:                <span v-if="careHistory.regId===undefined" style="color:red">选择患者后显示</span>                <span v-else style="color:red">{{ careHistory.regId }}</span>                <br>                病历编号:                <span v-if="careHistory.chId===undefined" style="color:red">保存病例后显示</span>                <span v-else style="color:blue">{{ careHistory.chId }}</span>              </el-col>              <el-col :span="12" style="text-align:right">                <el-button type="primary" :disabled="careHistory.regId===undefined" icon="el-icon-check" @click="handelSaveCareHistory">保存病历</el-button>                <el-button type="danger" :disabled="careHistory.regId===undefined" icon="el-icon-finished">完成就诊</el-button>              </el-col>            </el-row>          </el-card>          <el-card style="margin-top:3px">            <el-tabs :v-model="careActiveName" type="card">              <el-tab-pane label="病历">                <!-- 病例表单开始 -->                <el-card>                  <el-form ref="form" :model="careHistory" :inline="true" label-width="88px">                    <el-form-item label="发病日期" prop="caseDateObj">                      <el-date-picker                        v-model="caseDateObj"                        value-format="yyyy-MM-dd"                        size="small"                      />                    </el-form-item>                    <el-form-item label="接诊类型" prop="receiveType">                      <el-select                        v-model="careHistory.receiveType"                        placeholder="接诊类型"                        size="small"                      >                        <el-option                          v-for="dict in receiveTypeOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                    <el-form-item label="是否传染" prop="isContagious">                      <el-select                        v-model="careHistory.isContagious"                        placeholder="是否传染"                        size="small"                      >                        <el-option                          v-for="dict in isContagiousOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                  </el-form>                  <el-form ref="form" :model="careHistory" label-width="88px">                    <el-form-item label="主诉" :inline="false" prop="caseTitle">                      <el-input                        v-model="careHistory.caseTitle"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入主诉"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="诊断信息" :inline="false" prop="caseResult">                      <el-input                        v-model="careHistory.caseResult"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入诊断信息"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="医生建议" :inline="false" prop="doctorTips">                      <el-input                        v-model="careHistory.doctorTips"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入医生建议"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="备注" :inline="false" prop="remark">                      <el-input                        v-model="careHistory.remark"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入备注"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                  </el-form>                </el-card>                <!-- 病例表单结束 -->              </el-tab-pane>              <el-tab-pane label="处方">                <!-- 处方详情开始 -->                <el-collapse v-if="careOrders.length>0" accordion>                  <el-collapse-item v-for="(item,index) in careOrders" :key="index">                    <template slot="title">                      【{{ item.careOrder.coType==='0'?'药用处方':'检查处方' }}】【{{ index+1 }}】【处方总额】:￥{{ item.careOrder.allAmount }}                    </template>                    <el-table                      v-loading="loading"                      border                      :data="item.careOrderItems"                    >                      <el-table-column label="序号" align="center" type="index" width="50" />                      <el-table-column :label="item.careOrder.coType==='0'?'药品名称':'项目名称'" align="center" prop="itemName" />                      <el-table-column label="数量" align="center" prop="num" />                      <el-table-column label="单价(元)" align="center" prop="price" />                      <el-table-column label="金额(元)" align="center" prop="amount" />                      <el-table-column label="备注" align="center" prop="remark" />                      <el-table-column label="状态" align="center" prop="status" :formatter="orderDetailsStatusFormatter" />                      <el-table-column label="操作" align="center">                        <template slot-scope="scope">                          <el-button v-show="scope.row.status==='0'" type="text" icon="el-icon-delete" size="mini" @click="handleCareOrderItemDelete(scope.row)">删除</el-button>                        </template>                      </el-table-column>                    </el-table>                  </el-collapse-item>                </el-collapse>                <!-- 处方详情结束 -->                <div style="margin:3px;text-align:left">                  <el-button type="success" icon="el-icon-plus" @click="handelAddMedicinesOrder">添加药用处方</el-button>                  <el-button type="success" icon="el-icon-plus" @click="handelAddCheckItemOrder">添加检查处方</el-button>                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 右边的病历，处方结束 -->        </el-col>      </el-row>      <!-- 下面的整体卡片结束 -->      <!-- 弹出选择患者的层开始 -->      <el-dialog        title="请选择挂号患者"        :visible.sync="openRegistration"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <el-tabs v-model="activeName" :stretch="true" type="card" @tab-click="handleRegistrationTabClick">          <el-tab-pane label="待就诊列表" name="t1">            <el-table v-loading="tableLoading" border :data="toBeSeenRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleVisit(scope.row)">接诊</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊中列表" name="t2">            <el-table v-loading="tableLoading" border :data="visitingRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊完成列表" name="t3">            <el-table v-loading="tableLoading" border :data="visitCompletedRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>        </el-tabs>      </el-dialog>      <!-- 弹出选择患者的层结束 -->    </div>  </template>  <script>  import { queryToBeSeenRegistration, queryVisitingRegistration, queryVisitCompletedRegistration,    receivePatient, getPatientAllMessageByPatientId,    saveCareHistory, getCareHistoryByRegId, queryCareOrdersByChId } from '@/api/docter/care'  export default {    data() {      return {        // 总的遮罩层        loading: false,        // 表格的遮罩层        tableLoading: false,        // 门诊急诊的默认值        schedulingType: '1',        // 就诊类型的字典数据        schedulingTypeOptions: [],        // 排班类型的字典数据        subsectionTypeOptions: [],        // 性别数据字典        sexOptions: [],        // 接诊类型字典        receiveTypeOptions: [],        // 是否传染字典        isContagiousOptions: [],        // 处方详情的状态字典数据        orderDetailsStatusOptions: [],        // 是否打开选择挂号患者的弹层        openRegistration: false,        // 当前选中的挂号选项卡的        activeName: 't1',        // 右边的病例和处方的选项卡        careActiveName: 'c1',        // 待就诊数据        toBeSeenRegistration: [],        // 就诊中的数据        visitingRegistration: [],        // 就诊完成的数据        visitCompletedRegistration: [],        // 发病日期        caseDateObj: new Date(),        // 左边患者数据        patientAllObj: {          patientObj: {},          patientFileObj: {},          careHistoryObjList: []        },        // 提交到数据库的对象        careHistory: {          // 当前就诊中的挂号单ID          regId: undefined,          chId: undefined,          caseDate: undefined,          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        },        // 存放处方及详情数据        careOrders: []      }    },    created() {      // 加载排班类型 1 门诊  2急诊      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 排班时段 1 2 3 上午 下午 晚上      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 加载性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })      //  加载接诊类型字典      this.getDataByType('his\_receive\_type').then(res => {        this.receiveTypeOptions = res.data      })      //  加载是否传染字典      this.getDataByType('his\_contagious\_status').then(res => {        this.isContagiousOptions = res.data      })      //  加载处方详情的状态字典数据      this.getDataByType('his\_order\_details\_status').then(res => {        this.orderDetailsStatusOptions = res.data      })    },    methods: {      // 门诊急诊切换事件      schedulingTypeChange(value) {        this.schedulingType = value      },      // 打开选择挂号患者的弹出层      viewRegistration() {        this.activeName = 't1'        this.openRegistration = true        this.queryToBeSeenRegistration()      },      // 挂号患者弹出层的选项卡change事件      handleRegistrationTabClick(tab, event) {        if (tab.name === 't1') {          this.queryToBeSeenRegistration()        } else if (tab.name === 't2') {          this.queryVisitingRegistration()        } else if (tab.name === 't3') {          this.queryVisitCompletedRegistration()        }      },      // 查询待就诊的数据      queryToBeSeenRegistration() {        this.tableLoading = true        queryToBeSeenRegistration(this.schedulingType).then(res => {          this.toBeSeenRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊中的数据      queryVisitingRegistration() {        this.tableLoading = true        queryVisitingRegistration(this.schedulingType).then(res => {          this.visitingRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊完成的数据      queryVisitCompletedRegistration() {        this.tableLoading = true        queryVisitCompletedRegistration(this.schedulingType).then(res => {          this.visitCompletedRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 翻译挂号类型      schedulingTypeFormatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType)      },      // 翻译处方详情状态      orderDetailsStatusFormatter(row) {        return this.selectDictLabel(this.orderDetailsStatusOptions, row.status)      },      // 接诊      handleVisit(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 接诊，更新挂号的状态为就诊中并写入接诊的医生信息        this.loading = true        receivePatient(this.careHistory.regId).then(res => {          this.loading = false          this.msgSuccess('接诊成功')          // 根据患者ID查询患者信息档案信息 病历信息          getPatientAllMessageByPatientId(patientId).then(res => {            this.patientAllObj.patientObj = res.data.patient            this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)            this.patientAllObj.patientFileObj = res.data.patientFile            this.patientAllObj.careHistoryObjList = res.data.careHistoryList            this.openRegistration = false            this.loading = false          }).catch(() => {            this.msgError('查询患者信息失败')            this.loading = false          })        }).catch(() => {          this.msgError('接诊失败')          this.loading = false        })      },      // 载入      handleLoading(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 根据患者ID查询患者信息档案信息 病历信息        getPatientAllMessageByPatientId(patientId).then(res => {          this.patientAllObj.patientObj = res.data.patient          this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)          this.patientAllObj.patientFileObj = res.data.patientFile          this.patientAllObj.careHistoryObjList = res.data.careHistoryList          this.openRegistration = false          this.loading = false        }).catch(() => {          this.msgError('载入失败')          this.loading = false        })        // 根据挂号单ID查询对应的病历信息        this.getCareHistoryByRegId(row.regId)      },      // 根据挂号单ID查询对应的病历信息      getCareHistoryByRegId(regId) {        this.loading = true        getCareHistoryByRegId(regId).then(res => {          if (res.data != null) {            this.careHistory = res.data            this.caseDateObj = res.data.careDate            // 如果有数据根据病历ID查询处方信息            this.getCareOrdersByChId(res.data.chId)          } else {            this.resetCareHistoryAndCareOrders(regId)          }          this.loading = false        }).catch(() => {          this.msgError('查询病历失败')          this.loading = false        })      },      // 根据病历ID查询处方信息      getCareOrdersByChId(chId) {        this.loading = true        queryCareOrdersByChId(chId).then(res => {          this.careOrders = res.data          this.loading = false        }).catch(() => {          this.msgError('查询详情失败')          this.loading = false        })      },      // 重置右边的病历和处方详情数据      resetCareHistoryAndCareOrders(regId) {        this.careHistory = {          // 当前就诊中的挂号单ID          regId: regId,          chId: undefined,          caseDate: undefined,          caseDateObj: new Date(),          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        }        this.careOrders = []      },      // 保存病历      handelSaveCareHistory() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        // 封装数据        this.careHistory.caseDate = this.moment(this.caseDateObj).format('YYYY-MM-DD')        this.careHistory.patientId = this.patientAllObj.patientObj.patientId        this.careHistory.patientName = this.patientAllObj.patientObj.name        this.loading = true        saveCareHistory(this.careHistory).then(res => {          this.msgSuccess('保存病历成功')          this.loading = false          this.careHistory.chId = res.data.chId        }).catch(() => {          this.msgError('保存病历失败')          this.loading = false        })      },      // 删除处方详情      handleCareOrderItemDelete(row) {      },      // 打开添加药用处方的弹出层      handelAddMedicinesOrder() {      },      // 打开添加检查处方的弹出层      handelAddCheckItemOrder() {      }    }  }  </script>  <style scoped>    .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }    .item{      font-size: 14px;      margin-bottom: 8px;    }  </style> |

# 新开就诊第三部分

## 修改src/views/doctor/care/index.vue静态页面1

|  |
| --- |
| <template>    <div v-loading="loading" class="app-container">      <!-- 选择门诊和急诊的选项开始 -->      <el-card style="margin-bottom:5px;text-align:center">        <el-radio-group v-model="schedulingType" @change="schedulingTypeChange">          <el-radio-button            v-for="dict in schedulingTypeOptions"            :key="dict.dictValue"            :label="dict.dictValue"            :value="dict.dictValue"          >            {{ dict.dictLabel }}          </el-radio-button>        </el-radio-group>      </el-card>      <!-- 选择门诊和急诊的选项结束 -->      <!-- 下面的整体卡片开始 -->      <el-row :gutter="5">        <el-col :span="7">          <!-- 左边的患者信息开始 -->          <el-card>            <el-form ref="form" :model="patientAllObj.patientObj" :inline="true" label-width="88px">              <el-form-item label="患者姓名" prop="name">                <el-input                  v-model="patientAllObj.patientObj.name"                  placeholder="请输入患者姓名"                  :disabled="true"                  style="width:100%"                  size="small"                >                  <el-button slot="append" icon="el-icon-user-solid" @click="viewRegistration" />                </el-input>              </el-form-item>              <el-form-item label="身份证号" prop="idCard">                <el-input                  v-model="patientAllObj.patientObj.idCard"                  placeholder="请输入身份证号"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者电话" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.phone"                  placeholder="请输入患者电话"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="性别" prop="sex">                <el-radio-group v-model="patientAllObj.patientObj.sex">                  <el-radio                    v-for="d in sexOptions"                    :key="d.dictValue"                    :label="d.dictValue"                    :value="d.dictValue"                  >{{ d.dictLabel }}</el-radio>                </el-radio-group>              </el-form-item>              <el-form-item label="出生日期" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.birthday"                  placeholder="请选择出生日期"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者年龄" prop="age">                <el-input                  v-model="patientAllObj.patientObj.age"                  placeholder="请输入患者年龄"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者地址" prop="address">                <el-input                  v-model="patientAllObj.patientObj.address"                  placeholder="请输入患者地址"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="过敏信息" prop="allergyInfo">                <el-input                  v-model="patientAllObj.patientObj.allergyInfo"                  placeholder="请输入过敏信息"                  type="textarea"                  :autosize="{minRows:2,maxRows:4}"                  :disabled="true"                  maxlength="100"                  show-word-limit                  style="width:100%"                  size="small"                />              </el-form-item>            </el-form>            <el-tabs :stretch="true" type="card">              <el-tab-pane label="患者档案">                <div v-if="patientAllObj.patientObj.isFinal==='1'">                  <div class="item">                    紧急联系人:<span v-text="patientAllObj.patientFileObj.emergencyContactName" />                  </div>                  <div class="item">                    紧急联系人电话:<span v-text="patientAllObj.patientFileObj.emergencyContactPhone" />                  </div>                  <div class="item">                    关系: <span v-text="patientAllObj.patientFileObj.emergencyContactRelation" />                  </div>                  <div class="item">                    左耳听力:<span v-text="patientAllObj.patientFileObj.leftEarHearing" />                  </div>                  <div class="item">                    右耳听力:<span v-text="patientAllObj.patientFileObj.rightEarHearing" />                  </div>                  <div class="item">                    左眼视力:<span v-text="patientAllObj.patientFileObj.leftVision" />                  </div>                  <div class="item">                    右眼视力:<span v-text="patientAllObj.patientFileObj.rightVision" />                  </div>                  <div class="item">                    身高:<span v-text="patientAllObj.patientFileObj.height" />                  </div>                  <div class="item">                    体重:<span v-text="patientAllObj.patientFileObj.weight" />                  </div>                  <div class="item">                    血型:<span v-text="patientAllObj.patientFileObj.bloodType" />                  </div>                  <div class="item">                    个人史:<span v-text="patientAllObj.patientFileObj.personalInfo" />                  </div>                  <div class="item">                    家族史:<span v-text="patientAllObj.patientFileObj.familyInfo" />                  </div>                  <div class="item">                    档案创建时间:<span v-text="patientAllObj.patientFileObj.createTime" />                  </div>                  <div class="item">                    档案更新时间:<span v-text="patientAllObj.patientFileObj.updateTime" />                  </div>                </div>                <div v-else style="text-align:center">                  暂无患者档案信息                </div>              </el-tab-pane>              <el-tab-pane label="历史病历">                <div v-if="patientAllObj.careHistoryObjList.length>0">                  <el-collapse accordion>                    <el-collapse-item v-for="(item,index) in patientAllObj.careHistoryObjList" :key="index">                      <template slot="title">                        就诊部门:【{{ item.deptName }}】 就诊时间:{{ item.careDate }}                      </template>                      <div class="item">                        主诉:{{ item.caseTitle }}                      </div>                      <div class="item">                        发病日期:{{ item.caseDate }}                      </div>                      <div class="item">                        诊断信息:{{ item.caseResult }}                      </div>                      <div class="item">                        医生建议:{{ item.doctorTips }}                      </div>                      <div class="item">                        备注:{{ item.remark }}                      </div>                      <div class="item">                        接诊类型:{{ item.receiveType==='0'?'初诊':'复诊' }}                      </div>                      <div class="item">                        是否传染:{{ item.isContagious==='0'?'否':'是' }}                      </div>                    </el-collapse-item>                  </el-collapse>                </div>                <div v-else style="text-align:center">                  暂无患者病历信息                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 左边的患者信息结束 -->        </el-col>        <el-col :span="17">          <!-- 右边的病历，处方开始 -->          <el-card>            <el-row :gutter="5">              <el-col :span="12">                挂号单号:                <span v-if="careHistory.regId===undefined" style="color:red">选择患者后显示</span>                <span v-else style="color:red">{{ careHistory.regId }}</span>                <br>                病历编号:                <span v-if="careHistory.chId===undefined" style="color:red">保存病例后显示</span>                <span v-else style="color:blue">{{ careHistory.chId }}</span>              </el-col>              <el-col :span="12" style="text-align:right">                <el-button type="primary" :disabled="careHistory.regId===undefined" icon="el-icon-check" @click="handelSaveCareHistory">保存病历</el-button>                <el-button type="danger" :disabled="careHistory.regId===undefined" icon="el-icon-finished">完成就诊</el-button>              </el-col>            </el-row>          </el-card>          <el-card style="margin-top:3px">            <el-tabs :v-model="careActiveName" type="card">              <el-tab-pane label="病历">                <!-- 病例表单开始 -->                <el-card>                  <el-form ref="form" :model="careHistory" :inline="true" label-width="88px">                    <el-form-item label="发病日期" prop="caseDateObj">                      <el-date-picker                        v-model="caseDateObj"                        value-format="yyyy-MM-dd"                        size="small"                      />                    </el-form-item>                    <el-form-item label="接诊类型" prop="receiveType">                      <el-select                        v-model="careHistory.receiveType"                        placeholder="接诊类型"                        size="small"                      >                        <el-option                          v-for="dict in receiveTypeOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                    <el-form-item label="是否传染" prop="isContagious">                      <el-select                        v-model="careHistory.isContagious"                        placeholder="是否传染"                        size="small"                      >                        <el-option                          v-for="dict in isContagiousOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                  </el-form>                  <el-form ref="form" :model="careHistory" label-width="88px">                    <el-form-item label="主诉" :inline="false" prop="caseTitle">                      <el-input                        v-model="careHistory.caseTitle"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入主诉"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="诊断信息" :inline="false" prop="caseResult">                      <el-input                        v-model="careHistory.caseResult"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入诊断信息"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="医生建议" :inline="false" prop="doctorTips">                      <el-input                        v-model="careHistory.doctorTips"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入医生建议"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="备注" :inline="false" prop="remark">                      <el-input                        v-model="careHistory.remark"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入备注"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                  </el-form>                </el-card>                <!-- 病例表单结束 -->              </el-tab-pane>              <el-tab-pane label="处方">                <!-- 处方详情开始 -->                <el-collapse v-if="careOrders.length>0" accordion>                  <el-collapse-item v-for="(item,index) in careOrders" :key="index">                    <template slot="title">                      【{{ item.careOrder.coType==='0'?'药用处方':'检查处方' }}】【{{ index+1 }}】【处方总额】:￥{{ item.careOrder.allAmount }}                    </template>                    <el-table                      v-loading="loading"                      border                      :data="item.careOrderItems"                    >                      <el-table-column label="序号" align="center" type="index" width="50" />                      <el-table-column :label="item.careOrder.coType==='0'?'药品名称':'项目名称'" align="center" prop="itemName" />                      <el-table-column label="数量" align="center" prop="num" />                      <el-table-column label="单价(元)" align="center" prop="price" />                      <el-table-column label="金额(元)" align="center" prop="amount" />                      <el-table-column label="备注" align="center" prop="remark" />                      <el-table-column label="状态" align="center" prop="status" :formatter="orderDetailsStatusFormatter" />                      <el-table-column label="操作" align="center">                        <template slot-scope="scope">                          <el-button v-show="scope.row.status==='0'" type="text" icon="el-icon-delete" size="mini" @click="handleCareOrderItemDelete(scope.row)">删除</el-button>                        </template>                      </el-table-column>                    </el-table>                  </el-collapse-item>                </el-collapse>                <!-- 处方详情结束 -->                <div style="margin:3px;text-align:left">                  <el-button type="success" icon="el-icon-plus" :disabled="careHistory.chId===undefined" @click="handelAddMedicinesOrder">添加药用处方</el-button>                  <el-button type="success" icon="el-icon-plus" :disabled="careHistory.chId===undefined" @click="handelAddCheckItemOrder">添加检查处方</el-button>                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 右边的病历，处方结束 -->        </el-col>      </el-row>      <!-- 下面的整体卡片结束 -->      <!-- 弹出选择患者的层开始 -->      <el-dialog        title="请选择挂号患者"        :visible.sync="openRegistration"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <el-tabs v-model="activeName" :stretch="true" type="card" @tab-click="handleRegistrationTabClick">          <el-tab-pane label="待就诊列表" name="t1">            <el-table v-loading="tableLoading" border :data="toBeSeenRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleVisit(scope.row)">接诊</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊中列表" name="t2">            <el-table v-loading="tableLoading" border :data="visitingRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊完成列表" name="t3">            <el-table v-loading="tableLoading" border :data="visitCompletedRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>        </el-tabs>      </el-dialog>      <!-- 弹出选择患者的层结束 -->      <!-- 添加药用处方和检查处方的弹出层开始 -->      <el-dialog        :title="title"        :visible.sync="openAddOrderItem"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <div style="margin:3px;text-align:right">          【{{ submitCareOrder.careOrder.coType==='0'?'药用':'检查' }}】处方总额:￥<span style="color:red">{{ submitCareOrder.careOrder.allAmount }}</span>          <el-button style="margin-left:15px" type="success" icon="el-icon-plus" @click="handleSaveOrderItem">确定添加</el-button>        </div>        <el-table          border          :data="submitCareOrder.careOrderItems"        >          <el-table-column label="序号" align="center" type="index" width="50" />          <el-table-column :label="submitCareOrder.careOrder.coType==='0'?'药品名称':'项目名称'" align="center" prop="itemName" />          <el-table-column label="数量" align="center" prop="num">            <template slot-scope="scope">              <el-input-number                v-model="scope.row.num"                size="small"              />            </template>          </el-table-column>          <el-table-column label="单价(元)" align="center" prop="price" />          <el-table-column label="金额(元)" align="center" prop="amount" />          <el-table-column label="备注" align="center" prop="remark">            <template slot-scope="scope">              <el-input                v-model="scope.row.remark"                size="small"              />            </template>          </el-table-column>          <el-table-column label="操作" align="center">            <template slot-scope="scope">              <el-button type="text" icon="el-icon-delete" size="mini" @click="handleCareOrderItemDelete(scope.row)">删除</el-button>            </template>          </el-table-column>        </el-table>        <el-button type="primary" size="mini" style="width:100%;margin-top:2px" @click="handleOpenAddOrderItemDrawer">添加【{{ submitCareOrder.careOrder.coType==='0'?'药品':'检查' }}】项 </el-button>      </el-dialog>      <!-- 添加药用处方和检查出方的弹出层结束 -->      <!-- 药品列表的抽屉开始 -->      <el-drawer        :visible.sync="openDrawerMedicines"        direction="rtl"        append-to-body      >        <h2 align="center">药品列表</h2>        <el-form ref="queryItemForm" :model="queryItemFormParams" label-width="68px">          <el-form-item label="关键字" prop="keywords">            <el-input              v-model="queryItemFormParams.keywords"              placeholder="请输入关键字"              clearable              size="small"              style="width:180px"            />            <el-button type="primary" icon="el-icon-search" size="mini" @click="handleMedicinesFormQuery">查询</el-button>            <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetItemFormQuery">重置</el-button>          </el-form-item>          <el-table            v-loading="drawerLoading"            height="600px"            border            :data="tableItemList"            @selection-change="handleMedicinesSelectionChange"          >            <el-table-column type="selection" width="55" align="center" />            <el-table-column label="药品ID" align="center" prop="medicinesId" />            <el-table-column label="药品名称" align="center" prop="medicinesName" />            <el-table-column label="关键字" align="center" prop="keywords" />            <el-table-column label="库存量" align="center" prop="medicinesStockNum" />            <el-table-column label="单位" align="center">              <template slot-scope="scope">                {{ scope.row.conversion }}/ {{ scope.row.unit }}              </template>            </el-table-column>          </el-table>          <!-- 数据表格结束 -->          <!-- 分页控件开始 -->          <el-pagination            v-show="total>0"            :current-page="queryItemFormParams.pageNum"            :page-sizes="[5, 10, 20, 30]"            :page-size="queryItemFormParams.pageSize"            layout="total, sizes, prev, pager, next, jumper"            :total="total"            @size-change="handleMedicinesSizeChange"            @current-change="handleMedicinesCurrentChange"          />          <el-form-item>            <div style="text-align:center">              <el-button type="primary" icon="el-icon-check" size="mini">添加并关闭</el-button>            </div>          </el-form-item>        </el-form>      <!-- 分页控件结束 -->      </el-drawer>      <!-- 药品列表的抽屉结束 -->      <!-- 检查项目的抽屉开始 -->      <el-drawer        :visible.sync="openDrawerCheckItem"        direction="rtl"        append-to-body      >        <h2 align="center">检查项目列表</h2>        <el-form ref="queryItemForm" :model="queryItemFormParams" label-width="68px">          <el-form-item label="关键字" prop="keywords">            <el-input              v-model="queryItemFormParams.keywords"              placeholder="请输入关键字"              clearable              size="small"              style="width:180px"            />            <el-button type="primary" icon="el-icon-search" size="mini" @click="handleCheckItemFormQuery">查询</el-button>            <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetItemFormQuery">重置</el-button>          </el-form-item>          <el-table            v-loading="drawerLoading"            height="600px"            border            :data="tableItemList"            @selection-change="handleCheckItemSelectionChange"          >            <el-table-column type="selection" width="55" align="center" />            <el-table-column label="项目ID" align="center" prop="checkItemId" />            <el-table-column label="项目名称" align="center" prop="checkItemName" />            <el-table-column label="关键字" align="center" prop="keywords" />            <el-table-column label="项目单价" align="center" prop="unitPrice" />            <el-table-column label="单位" align="center" prop="unit" />          </el-table>          <!-- 数据表格结束 -->          <!-- 分页控件开始 -->          <el-pagination            v-show="total>0"            :current-page="queryItemFormParams.pageNum"            :page-sizes="[5, 10, 20, 30]"            :page-size="queryItemFormParams.pageSize"            layout="total, sizes, prev, pager, next, jumper"            :total="total"            @size-change="handleCheckItemSizeChange"            @current-change="handleCheckItemCurrentChange"          />          <el-form-item>            <div style="text-align:center">              <el-button type="primary" icon="el-icon-check" size="mini">添加并关闭</el-button>            </div>          </el-form-item>        </el-form>      </el-drawer>      <!-- 检查项目的抽屉结束 -->    </div>  </template>  <script>  import { queryToBeSeenRegistration, queryVisitingRegistration, queryVisitCompletedRegistration,    receivePatient, getPatientAllMessageByPatientId,    saveCareHistory, getCareHistoryByRegId, queryCareOrdersByChId } from '@/api/docter/care'  import { listCheckItemForPage } from '@/api/system/checkItem'  import { listMedicinesForPage } from '@/api/erp/medicines'  export default {    data() {      return {        // 总的遮罩层        loading: false,        // 表格的遮罩层        tableLoading: false,        // 门诊急诊的默认值        schedulingType: '1',        // 就诊类型的字典数据        schedulingTypeOptions: [],        // 排班类型的字典数据        subsectionTypeOptions: [],        // 性别数据字典        sexOptions: [],        // 接诊类型字典        receiveTypeOptions: [],        // 是否传染字典        isContagiousOptions: [],        // 处方详情的状态字典数据        orderDetailsStatusOptions: [],        // 是否打开选择挂号患者的弹层        openRegistration: false,        // 当前选中的挂号选项卡的        activeName: 't1',        // 右边的病例和处方的选项卡        careActiveName: 'c1',        // 待就诊数据        toBeSeenRegistration: [],        // 就诊中的数据        visitingRegistration: [],        // 就诊完成的数据        visitCompletedRegistration: [],        // 发病日期        caseDateObj: new Date(),        // 左边患者数据        patientAllObj: {          patientObj: {},          patientFileObj: {},          careHistoryObjList: []        },        // 提交到数据库的对象        careHistory: {          // 当前就诊中的挂号单ID          regId: undefined,          chId: undefined,          caseDate: undefined,          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        },        // 存放处方及详情数据        careOrders: [],        // 最后要提交到的处方及详情        submitCareOrder: {          careOrder: {            allAmount: 0.00,            patientId: undefined,            patientName: undefined,            coType: '0' // 默认为药用处方          },          careOrderItems: [            { itemName: '111',              num: 10,              price: 10,              amount: 100,              remark: 'aaaa' }          ]        },        // 弹出层的标题        title: '',        // 是否打开药品和检查项的添加的弹出层        openAddOrderItem: false,        // 是否打开药品列表的抽屉        openDrawerMedicines: false,        // 是否打开检查项目列表的抽屉        openDrawerCheckItem: false,        // 药品和检查项目抽屉的查询条件        queryItemFormParams: {          pageNum: 1,          pageSize: 10,          keywords: undefined        },        // 药品或检查项目表格的数据        tableItemList: [],        // 抽屉里面数据的总条数        total: 0,        // 抽屉数据加载的遮罩        drawerLoading: false,        // 抽屉里面选中的数据        selectItemList: []      }    },    created() {      // 加载排班类型 1 门诊  2急诊      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 排班时段 1 2 3 上午 下午 晚上      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 加载性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })      //  加载接诊类型字典      this.getDataByType('his\_receive\_type').then(res => {        this.receiveTypeOptions = res.data      })      //  加载是否传染字典      this.getDataByType('his\_contagious\_status').then(res => {        this.isContagiousOptions = res.data      })      //  加载处方详情的状态字典数据      this.getDataByType('his\_order\_details\_status').then(res => {        this.orderDetailsStatusOptions = res.data      })    },    methods: {      // 门诊急诊切换事件      schedulingTypeChange(value) {        this.schedulingType = value      },      // 打开选择挂号患者的弹出层      viewRegistration() {        this.activeName = 't1'        this.openRegistration = true        this.queryToBeSeenRegistration()      },      // 挂号患者弹出层的选项卡change事件      handleRegistrationTabClick(tab, event) {        if (tab.name === 't1') {          this.queryToBeSeenRegistration()        } else if (tab.name === 't2') {          this.queryVisitingRegistration()        } else if (tab.name === 't3') {          this.queryVisitCompletedRegistration()        }      },      // 查询待就诊的数据      queryToBeSeenRegistration() {        this.tableLoading = true        queryToBeSeenRegistration(this.schedulingType).then(res => {          this.toBeSeenRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊中的数据      queryVisitingRegistration() {        this.tableLoading = true        queryVisitingRegistration(this.schedulingType).then(res => {          this.visitingRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊完成的数据      queryVisitCompletedRegistration() {        this.tableLoading = true        queryVisitCompletedRegistration(this.schedulingType).then(res => {          this.visitCompletedRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 翻译挂号类型      schedulingTypeFormatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType)      },      // 翻译处方详情状态      orderDetailsStatusFormatter(row) {        return this.selectDictLabel(this.orderDetailsStatusOptions, row.status)      },      // 接诊      handleVisit(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 接诊，更新挂号的状态为就诊中并写入接诊的医生信息        this.loading = true        receivePatient(this.careHistory.regId).then(res => {          this.loading = false          this.msgSuccess('接诊成功')          // 根据患者ID查询患者信息档案信息 病历信息          getPatientAllMessageByPatientId(patientId).then(res => {            this.patientAllObj.patientObj = res.data.patient            this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)            this.patientAllObj.patientFileObj = res.data.patientFile            this.patientAllObj.careHistoryObjList = res.data.careHistoryList            this.openRegistration = false            this.loading = false          }).catch(() => {            this.msgError('查询患者信息失败')            this.loading = false          })        }).catch(() => {          this.msgError('接诊失败')          this.loading = false        })      },      // 载入      handleLoading(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 根据患者ID查询患者信息档案信息 病历信息        getPatientAllMessageByPatientId(patientId).then(res => {          this.patientAllObj.patientObj = res.data.patient          this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)          this.patientAllObj.patientFileObj = res.data.patientFile          this.patientAllObj.careHistoryObjList = res.data.careHistoryList          this.openRegistration = false          this.loading = false          // 查询当前挂号单之前对应的病例信息        }).catch(() => {          this.msgError('载入失败')          this.loading = false        })        // 根据挂号单ID查询对应的病历信息        this.getCareHistoryByRegId(row.regId)      },      // 根据挂号单ID查询对应的病历信息      getCareHistoryByRegId(regId) {        this.loading = true        getCareHistoryByRegId(regId).then(res => {          if (res.data != null) {            this.careHistory = res.data            this.caseDateObj = res.data.careDate            // 如果有数据根据病历ID查询处方信息            this.getCareOrdersByChId(res.data.chId)          } else {            this.resetCareHistoryAndCareOrders(regId)          }          this.loading = false        }).catch(() => {          this.msgError('查询病历失败')          this.loading = false        })      },      // 根据病历ID查询处方信息      getCareOrdersByChId(chId) {        this.loading = true        queryCareOrdersByChId(chId).then(res => {          this.careOrders = res.data          this.loading = false        }).catch(() => {          this.msgError('查询详情失败')          this.loading = false        })      },      // 重置右边的病历和处方详情数据      resetCareHistoryAndCareOrders(regId) {        this.careHistory = {          // 当前就诊中的挂号单ID          regId: regId,          chId: undefined,          caseDate: undefined,          caseDateObj: new Date(),          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        }        this.careOrders = []      },      // 保存病历      handelSaveCareHistory() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        // 封装数据        this.careHistory.caseDate = this.moment(this.caseDateObj).format('YYYY-MM-DD')        this.careHistory.patientId = this.patientAllObj.patientObj.patientId        this.careHistory.patientName = this.patientAllObj.patientObj.name        this.loading = true        saveCareHistory(this.careHistory).then(res => {          this.msgSuccess('保存病历成功')          this.loading = false          this.careHistory.chId = res.data.chId        }).catch(() => {          this.msgError('保存病历失败')          this.loading = false        })      },      // 删除处方详情      handleCareOrderItemDelete(row) {      },      // 打开添加药用处方的弹出层      handelAddMedicinesOrder() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        if (!this.careHistory.chId) {          this.msgError('请先保存病历')          return        }        this.submitCareOrder.careOrder.coType = '0'        this.title = '添加【药用】处方'        this.openAddOrderItem = true      },      // 打开添加检查处方的弹出层      handelAddCheckItemOrder() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        if (!this.careHistory.chId) {          this.msgError('请先保存病历')          return        }        this.submitCareOrder.careOrder.coType = '1'        this.title = '添加【检查】处方'        this.openAddOrderItem = true      },      // 打开药口或者检查项目的抽屉      handleOpenAddOrderItemDrawer() {        if (this.submitCareOrder.careOrder.coType === '0') {          // 打开药口列表抽屉          this.openDrawerMedicines = true          this.resetItemFormQuery()          this.getMedicinesList()        } else {          // 打开检查项目的抽屉          this.openDrawerCheckItem = true          this.resetItemFormQuery()          this.getCheckItemList()        }      },      // 加载药品数据      getMedicinesList() {        this.tableItemList = []        this.drawerLoading = true        listMedicinesForPage(this.queryItemFormParams).then(res => {          this.tableItemList = res.data          this.total = res.total          this.drawerLoading = false        }).catch(() => {          this.drawerLoading = false          this.msgError('查询药品失败')        })      },      // 数据表格的多选择框选择时触发      handleMedicinesSelectionChange(selection) {        this.selectItemList = selection      },      // 分页pageSize变化时触发      handleMedicinesSizeChange(val) {        this.queryItemFormParams.pageSize = val        // 重新查询        this.getMedicinesList()      },      // 点击上一页  下一页，跳转到哪一页面时触发      handleMedicinesCurrentChange(val) {        this.queryItemFormParams.pageNum = val        // 重新查询        this.getMedicinesList()      },      // 搜索药品的方法      handleMedicinesFormQuery() {        this.queryItemFormParams.pageNum = 1        this.getMedicinesList()      },      // 加载药品数据      getCheckItemList() {        this.tableItemList = []        this.drawerLoading = true        listCheckItemForPage(this.queryItemFormParams).then(res => {          this.tableItemList = res.data          this.total = res.total          this.drawerLoading = false        }).catch(() => {          this.drawerLoading = false          this.msgError('查询检查项目失败')        })      },      // 数据表格的多选择框选择时触发      handleCheckItemSelectionChange(selection) {        this.selectItemList = selection      },      // 分页pageSize变化时触发      handleCheckItemSizeChange(val) {        this.queryItemFormParams.pageSize = val        // 重新查询        this.getCheckItemList()      },      // 点击上一页  下一页，跳转到哪一页面时触发      handleCheckItemCurrentChange(val) {        this.queryItemFormParams.pageNum = val        // 重新查询        this.getCheckItemList()      },      // 搜索检查项目的方法      handleCheckItemFormQuery() {        this.queryItemFormParams.pageNum = 1        this.getCheckItemList()      },      // 重置抽屉的查询条件      resetItemFormQuery() {        this.queryItemFormParams = {          pageNum: 1,          pageSize: 10,          keywords: undefined        }        if (this.submitCareOrder.careOrder.coType === '0') {          this.getMedicinesList()        } else {          this.getCheckItemList()        }      },      // 保存处方单信息      handleSaveOrderItem() {      }    }  }  </script>  <style scoped>    .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }    .item{      font-size: 14px;      margin-bottom: 8px;    }  </style> |

## 修改src/views/doctor/care/index.vue静态页面2

|  |
| --- |
| <template>    <div v-loading="loading" class="app-container">      <!-- 选择门诊和急诊的选项开始 -->      <el-card style="margin-bottom:5px;text-align:center">        <el-radio-group v-model="schedulingType" @change="schedulingTypeChange">          <el-radio-button            v-for="dict in schedulingTypeOptions"            :key="dict.dictValue"            :label="dict.dictValue"            :value="dict.dictValue"          >            {{ dict.dictLabel }}          </el-radio-button>        </el-radio-group>      </el-card>      <!-- 选择门诊和急诊的选项结束 -->      <!-- 下面的整体卡片开始 -->      <el-row :gutter="5">        <el-col :span="7">          <!-- 左边的患者信息开始 -->          <el-card>            <el-form ref="form" :model="patientAllObj.patientObj" :inline="true" label-width="88px">              <el-form-item label="患者姓名" prop="name">                <el-input                  v-model="patientAllObj.patientObj.name"                  placeholder="请输入患者姓名"                  :disabled="true"                  style="width:100%"                  size="small"                >                  <el-button slot="append" icon="el-icon-user-solid" @click="viewRegistration" />                </el-input>              </el-form-item>              <el-form-item label="身份证号" prop="idCard">                <el-input                  v-model="patientAllObj.patientObj.idCard"                  placeholder="请输入身份证号"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者电话" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.phone"                  placeholder="请输入患者电话"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="性别" prop="sex">                <el-radio-group v-model="patientAllObj.patientObj.sex">                  <el-radio                    v-for="d in sexOptions"                    :key="d.dictValue"                    :label="d.dictValue"                    :value="d.dictValue"                  >{{ d.dictLabel }}</el-radio>                </el-radio-group>              </el-form-item>              <el-form-item label="出生日期" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.birthday"                  placeholder="请选择出生日期"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者年龄" prop="age">                <el-input                  v-model="patientAllObj.patientObj.age"                  placeholder="请输入患者年龄"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者地址" prop="address">                <el-input                  v-model="patientAllObj.patientObj.address"                  placeholder="请输入患者地址"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="过敏信息" prop="allergyInfo">                <el-input                  v-model="patientAllObj.patientObj.allergyInfo"                  placeholder="请输入过敏信息"                  type="textarea"                  :autosize="{minRows:2,maxRows:4}"                  :disabled="true"                  maxlength="100"                  show-word-limit                  style="width:100%"                  size="small"                />              </el-form-item>            </el-form>            <el-tabs :stretch="true" type="card">              <el-tab-pane label="患者档案">                <div v-if="patientAllObj.patientObj.isFinal==='1'">                  <div class="item">                    紧急联系人:<span v-text="patientAllObj.patientFileObj.emergencyContactName" />                  </div>                  <div class="item">                    紧急联系人电话:<span v-text="patientAllObj.patientFileObj.emergencyContactPhone" />                  </div>                  <div class="item">                    关系: <span v-text="patientAllObj.patientFileObj.emergencyContactRelation" />                  </div>                  <div class="item">                    左耳听力:<span v-text="patientAllObj.patientFileObj.leftEarHearing" />                  </div>                  <div class="item">                    右耳听力:<span v-text="patientAllObj.patientFileObj.rightEarHearing" />                  </div>                  <div class="item">                    左眼视力:<span v-text="patientAllObj.patientFileObj.leftVision" />                  </div>                  <div class="item">                    右眼视力:<span v-text="patientAllObj.patientFileObj.rightVision" />                  </div>                  <div class="item">                    身高:<span v-text="patientAllObj.patientFileObj.height" />                  </div>                  <div class="item">                    体重:<span v-text="patientAllObj.patientFileObj.weight" />                  </div>                  <div class="item">                    血型:<span v-text="patientAllObj.patientFileObj.bloodType" />                  </div>                  <div class="item">                    个人史:<span v-text="patientAllObj.patientFileObj.personalInfo" />                  </div>                  <div class="item">                    家族史:<span v-text="patientAllObj.patientFileObj.familyInfo" />                  </div>                  <div class="item">                    档案创建时间:<span v-text="patientAllObj.patientFileObj.createTime" />                  </div>                  <div class="item">                    档案更新时间:<span v-text="patientAllObj.patientFileObj.updateTime" />                  </div>                </div>                <div v-else style="text-align:center">                  暂无患者档案信息                </div>              </el-tab-pane>              <el-tab-pane label="历史病历">                <div v-if="patientAllObj.careHistoryObjList.length>0">                  <el-collapse accordion>                    <el-collapse-item v-for="(item,index) in patientAllObj.careHistoryObjList" :key="index">                      <template slot="title">                        就诊部门:【{{ item.deptName }}】 就诊时间:{{ item.careDate }}                      </template>                      <div class="item">                        主诉:{{ item.caseTitle }}                      </div>                      <div class="item">                        发病日期:{{ item.caseDate }}                      </div>                      <div class="item">                        诊断信息:{{ item.caseResult }}                      </div>                      <div class="item">                        医生建议:{{ item.doctorTips }}                      </div>                      <div class="item">                        备注:{{ item.remark }}                      </div>                      <div class="item">                        接诊类型:{{ item.receiveType==='0'?'初诊':'复诊' }}                      </div>                      <div class="item">                        是否传染:{{ item.isContagious==='0'?'否':'是' }}                      </div>                    </el-collapse-item>                  </el-collapse>                </div>                <div v-else style="text-align:center">                  暂无患者病历信息                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 左边的患者信息结束 -->        </el-col>        <el-col :span="17">          <!-- 右边的病历，处方开始 -->          <el-card>            <el-row :gutter="5">              <el-col :span="12">                挂号单号:                <span v-if="careHistory.regId===undefined" style="color:red">选择患者后显示</span>                <span v-else style="color:red">{{ careHistory.regId }}</span>                <br>                病历编号:                <span v-if="careHistory.chId===undefined" style="color:red">保存病例后显示</span>                <span v-else style="color:blue">{{ careHistory.chId }}</span>              </el-col>              <el-col :span="12" style="text-align:right">                <el-button type="primary" :disabled="careHistory.regId===undefined" icon="el-icon-check" @click="handelSaveCareHistory">保存病历</el-button>                <el-button type="danger" :disabled="careHistory.regId===undefined" icon="el-icon-finished">完成就诊</el-button>              </el-col>            </el-row>          </el-card>          <el-card style="margin-top:3px">            <el-tabs :v-model="careActiveName" type="card">              <el-tab-pane label="病历">                <!-- 病例表单开始 -->                <el-card>                  <el-form ref="form" :model="careHistory" :inline="true" label-width="88px">                    <el-form-item label="发病日期" prop="caseDateObj">                      <el-date-picker                        v-model="caseDateObj"                        value-format="yyyy-MM-dd"                        size="small"                      />                    </el-form-item>                    <el-form-item label="接诊类型" prop="receiveType">                      <el-select                        v-model="careHistory.receiveType"                        placeholder="接诊类型"                        size="small"                      >                        <el-option                          v-for="dict in receiveTypeOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                    <el-form-item label="是否传染" prop="isContagious">                      <el-select                        v-model="careHistory.isContagious"                        placeholder="是否传染"                        size="small"                      >                        <el-option                          v-for="dict in isContagiousOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                  </el-form>                  <el-form ref="form" :model="careHistory" label-width="88px">                    <el-form-item label="主诉" :inline="false" prop="caseTitle">                      <el-input                        v-model="careHistory.caseTitle"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入主诉"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="诊断信息" :inline="false" prop="caseResult">                      <el-input                        v-model="careHistory.caseResult"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入诊断信息"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="医生建议" :inline="false" prop="doctorTips">                      <el-input                        v-model="careHistory.doctorTips"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入医生建议"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="备注" :inline="false" prop="remark">                      <el-input                        v-model="careHistory.remark"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入备注"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                  </el-form>                </el-card>                <!-- 病例表单结束 -->              </el-tab-pane>              <el-tab-pane label="处方">                <!-- 处方详情开始 -->                <el-collapse v-if="careOrders.length>0" accordion>                  <el-collapse-item v-for="(item,index) in careOrders" :key="index">                    <template slot="title">                      【{{ item.careOrder.coType==='0'?'药用处方':'检查处方' }}】【{{ index+1 }}】【处方总额】:￥{{ item.careOrder.allAmount }}                    </template>                    <el-table                      v-loading="loading"                      border                      :data="item.careOrderItems"                    >                      <el-table-column label="序号" align="center" type="index" width="50" />                      <el-table-column :label="item.careOrder.coType==='0'?'药品名称':'项目名称'" align="center" prop="itemName" />                      <el-table-column label="数量" align="center" prop="num" />                      <el-table-column label="单价(元)" align="center" prop="price" />                      <el-table-column label="金额(元)" align="center" prop="amount" />                      <el-table-column label="备注" align="center" prop="remark" />                      <el-table-column label="状态" align="center" prop="status" :formatter="orderDetailsStatusFormatter" />                      <el-table-column label="操作" align="center">                        <template slot-scope="scope">                          <el-button v-show="scope.row.status==='0'" type="text" icon="el-icon-delete" size="mini">删除</el-button>                        </template>                      </el-table-column>                    </el-table>                  </el-collapse-item>                </el-collapse>                <!-- 处方详情结束 -->                <div style="margin:3px;text-align:left">                  <el-button type="success" icon="el-icon-plus" :disabled="careHistory.chId===undefined" @click="handelAddMedicinesOrder">添加药用处方</el-button>                  <el-button type="success" icon="el-icon-plus" :disabled="careHistory.chId===undefined" @click="handelAddCheckItemOrder">添加检查处方</el-button>                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 右边的病历，处方结束 -->        </el-col>      </el-row>      <!-- 下面的整体卡片结束 -->      <!-- 弹出选择患者的层开始 -->      <el-dialog        title="请选择挂号患者"        :visible.sync="openRegistration"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <el-tabs v-model="activeName" :stretch="true" type="card" @tab-click="handleRegistrationTabClick">          <el-tab-pane label="待就诊列表" name="t1">            <el-table v-loading="tableLoading" border :data="toBeSeenRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleVisit(scope.row)">接诊</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊中列表" name="t2">            <el-table v-loading="tableLoading" border :data="visitingRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊完成列表" name="t3">            <el-table v-loading="tableLoading" border :data="visitCompletedRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>        </el-tabs>      </el-dialog>      <!-- 弹出选择患者的层结束 -->      <!-- 添加药用处方和检查出方的弹出层开始 -->      <el-dialog        :title="title"        :visible.sync="openAddOrderItem"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <div style="margin:3px;text-align:right">          【{{ submitCareOrder.careOrder.coType==='0'?'药用':'检查' }}】处方总额:￥<span style="color:red">{{ submitCareOrder.careOrder.allAmount }}</span>          <el-button style="margin-left:15px" type="success" icon="el-icon-plus" @click="handleSaveOrderItem">确定添加</el-button>        </div>        <el-table          border          :data="submitCareOrder.careOrderItems"          :row-class-name="tableRowClassName"        >          <el-table-column label="序号" align="center" type="index" width="50" />          <el-table-column :label="submitCareOrder.careOrder.coType==='0'?'药品名称':'项目名称'" align="center" prop="itemName" />          <el-table-column label="数量" width="180px" align="center" prop="num">            <template slot-scope="scope">              <el-input-number                v-model="scope.row.num"                size="small"                @change="handleCareOrderItemNumChange(scope.row)"              />            </template>          </el-table-column>          <el-table-column label="单价(元)" align="center" prop="price" />          <el-table-column label="金额(元)" align="center" prop="amount" />          <el-table-column label="备注" align="center" prop="remark">            <template slot-scope="scope">              <el-input                v-model="scope.row.remark"                size="small"              />            </template>          </el-table-column>          <el-table-column label="操作" align="center">            <template slot-scope="scope">              <el-button type="text" icon="el-icon-delete" size="mini" @click="handleCareOrderItemDelete(scope.row)">删除</el-button>            </template>          </el-table-column>        </el-table>        <el-button type="primary" size="mini" style="width:100%;margin-top:2px" @click="handleOpenAddOrderItemDrawer">添加【{{ submitCareOrder.careOrder.coType==='0'?'药品':'检查' }}】项 </el-button>      </el-dialog>      <!-- 添加药用处方和检查处方的弹出层结束 -->      <!-- 药品列表的抽屉开始 -->      <el-drawer        :visible.sync="openDrawerMedicines"        direction="rtl"        append-to-body      >        <h2 align="center">药品列表</h2>        <el-form ref="queryItemForm" :model="queryItemFormParams" label-width="68px">          <el-form-item label="关键字" prop="keywords">            <el-input              v-model="queryItemFormParams.keywords"              placeholder="请输入关键字"              clearable              size="small"              style="width:180px"            />            <el-button type="primary" icon="el-icon-search" size="mini" @click="handleMedicinesFormQuery">查询</el-button>            <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetItemFormQuery">重置</el-button>          </el-form-item>          <el-table            v-loading="drawerLoading"            height="600px"            border            :data="tableItemList"            @selection-change="handleMedicinesSelectionChange"          >            <el-table-column type="selection" width="55" align="center" />            <el-table-column label="药品ID" align="center" prop="medicinesId" />            <el-table-column label="药品名称" align="center" prop="medicinesName" />            <el-table-column label="关键字" align="center" prop="keywords" />            <el-table-column label="库存量" align="center" prop="medicinesStockNum" />            <el-table-column label="单位" align="center">              <template slot-scope="scope">                {{ scope.row.conversion }}/ {{ scope.row.unit }}              </template>            </el-table-column>          </el-table>          <!-- 数据表格结束 -->          <!-- 分页控件开始 -->          <el-pagination            v-show="total>0"            :current-page="queryItemFormParams.pageNum"            :page-sizes="[5, 10, 20, 30]"            :page-size="queryItemFormParams.pageSize"            layout="total, sizes, prev, pager, next, jumper"            :total="total"            @size-change="handleMedicinesSizeChange"            @current-change="handleMedicinesCurrentChange"          />          <el-form-item>            <div style="text-align:center">              <el-button type="primary" icon="el-icon-check" size="mini" @click="hanldeAddCareItem">添加并关闭</el-button>            </div>          </el-form-item>        </el-form>      <!-- 分页控件结束 -->      </el-drawer>      <!-- 药品列表的抽屉结束 -->      <!-- 检查项目的抽屉开始 -->      <el-drawer        :visible.sync="openDrawerCheckItem"        direction="rtl"        append-to-body      >        <h2 align="center">检查项目列表</h2>        <el-form ref="queryItemForm" :model="queryItemFormParams" label-width="68px">          <el-form-item label="关键字" prop="keywords">            <el-input              v-model="queryItemFormParams.keywords"              placeholder="请输入关键字"              clearable              size="small"              style="width:180px"            />            <el-button type="primary" icon="el-icon-search" size="mini" @click="handleCheckItemFormQuery">查询</el-button>            <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetItemFormQuery">重置</el-button>          </el-form-item>          <el-table            v-loading="drawerLoading"            height="600px"            border            :data="tableItemList"            @selection-change="handleCheckItemSelectionChange"          >            <el-table-column type="selection" width="55" align="center" />            <el-table-column label="项目ID" align="center" prop="checkItemId" />            <el-table-column label="项目名称" align="center" prop="checkItemName" />            <el-table-column label="关键字" align="center" prop="keywords" />            <el-table-column label="项目单价" align="center" prop="unitPrice" />            <el-table-column label="单位" align="center" prop="unit" />          </el-table>          <!-- 数据表格结束 -->          <!-- 分页控件开始 -->          <el-pagination            v-show="total>0"            :current-page="queryItemFormParams.pageNum"            :page-sizes="[5, 10, 20, 30]"            :page-size="queryItemFormParams.pageSize"            layout="total, sizes, prev, pager, next, jumper"            :total="total"            @size-change="handleCheckItemSizeChange"            @current-change="handleCheckItemCurrentChange"          />          <el-form-item>            <div style="text-align:center">              <el-button type="primary" icon="el-icon-check" size="mini" @click="hanldeAddCareItem">添加并关闭</el-button>            </div>          </el-form-item>        </el-form>      </el-drawer>      <!-- 检查项目的抽屉结束 -->    </div>  </template>  <script>  import { queryToBeSeenRegistration, queryVisitingRegistration, queryVisitCompletedRegistration,    receivePatient, getPatientAllMessageByPatientId,    saveCareHistory, getCareHistoryByRegId, queryCareOrdersByChId } from '@/api/docter/care'  import { listCheckItemForPage } from '@/api/system/checkItem'  import { listMedicinesForPage } from '@/api/erp/medicines'  export default {    data() {      return {        // 总的遮罩层        loading: false,        // 表格的遮罩层        tableLoading: false,        // 门诊急诊的默认值        schedulingType: '1',        // 就诊类型的字典数据        schedulingTypeOptions: [],        // 排班类型的字典数据        subsectionTypeOptions: [],        // 性别数据字典        sexOptions: [],        // 接诊类型字典        receiveTypeOptions: [],        // 是否传染字典        isContagiousOptions: [],        // 处方详情的状态字典数据        orderDetailsStatusOptions: [],        // 是否打开选择挂号患者的弹层        openRegistration: false,        // 当前选中的挂号选项卡的        activeName: 't1',        // 右边的病例和处方的选项卡        careActiveName: 'c1',        // 待就诊数据        toBeSeenRegistration: [],        // 就诊中的数据        visitingRegistration: [],        // 就诊完成的数据        visitCompletedRegistration: [],        // 发病日期        caseDateObj: new Date(),        // 左边患者数据        patientAllObj: {          patientObj: {},          patientFileObj: {},          careHistoryObjList: []        },        // 提交到数据库的对象        careHistory: {          // 当前就诊中的挂号单ID          regId: undefined,          chId: undefined,          caseDate: undefined,          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        },        // 存放处方及详情数据        careOrders: [],        // 最后要提交到的处方及详情        submitCareOrder: {          careOrder: {            allAmount: 0.00,            patientId: undefined,            patientName: undefined,            coType: '0' // 默认为药用处方          },          careOrderItems: []        },        // 弹出层的标题        title: '',        // 是否打开药品和检查项的添加的弹出层        openAddOrderItem: false,        // 是否打开药品列表的抽屉        openDrawerMedicines: false,        // 是否打开检查项目列表的抽屉        openDrawerCheckItem: false,        // 药品和检查项目抽屉的查询条件        queryItemFormParams: {          pageNum: 1,          pageSize: 10,          keywords: undefined        },        // 药品或检查项目表格的数据        tableItemList: [],        // 抽屉里面数据的总条数        total: 0,        // 抽屉数据加载的遮罩        drawerLoading: false,        // 抽屉里面选中的数据        selectItemList: []      }    },    created() {      // 加载排班类型 1 门诊  2急诊      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 排班时段 1 2 3 上午 下午 晚上      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 加载性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })      //  加载接诊类型字典      this.getDataByType('his\_receive\_type').then(res => {        this.receiveTypeOptions = res.data      })      //  加载是否传染字典      this.getDataByType('his\_contagious\_status').then(res => {        this.isContagiousOptions = res.data      })      //  加载处方详情的状态字典数据      this.getDataByType('his\_order\_details\_status').then(res => {        this.orderDetailsStatusOptions = res.data      })    },    methods: {      // 门诊急诊切换事件      schedulingTypeChange(value) {        this.schedulingType = value      },      // 打开选择挂号患者的弹出层      viewRegistration() {        this.activeName = 't1'        this.openRegistration = true        this.queryToBeSeenRegistration()      },      // 挂号患者弹出层的选项卡change事件      handleRegistrationTabClick(tab, event) {        if (tab.name === 't1') {          this.queryToBeSeenRegistration()        } else if (tab.name === 't2') {          this.queryVisitingRegistration()        } else if (tab.name === 't3') {          this.queryVisitCompletedRegistration()        }      },      // 查询待就诊的数据      queryToBeSeenRegistration() {        this.tableLoading = true        queryToBeSeenRegistration(this.schedulingType).then(res => {          this.toBeSeenRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊中的数据      queryVisitingRegistration() {        this.tableLoading = true        queryVisitingRegistration(this.schedulingType).then(res => {          this.visitingRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊完成的数据      queryVisitCompletedRegistration() {        this.tableLoading = true        queryVisitCompletedRegistration(this.schedulingType).then(res => {          this.visitCompletedRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 翻译挂号类型      schedulingTypeFormatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType)      },      // 翻译处方详情状态      orderDetailsStatusFormatter(row) {        return this.selectDictLabel(this.orderDetailsStatusOptions, row.status)      },      // 接诊      handleVisit(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 接诊，更新挂号的状态为就诊中并写入接诊的医生信息        this.loading = true        receivePatient(this.careHistory.regId).then(res => {          this.loading = false          this.msgSuccess('接诊成功')          // 根据患者ID查询患者信息档案信息 病历信息          getPatientAllMessageByPatientId(patientId).then(res => {            this.patientAllObj.patientObj = res.data.patient            this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)            this.patientAllObj.patientFileObj = res.data.patientFile            this.patientAllObj.careHistoryObjList = res.data.careHistoryList            this.openRegistration = false            this.loading = false          }).catch(() => {            this.msgError('查询患者信息失败')            this.loading = false          })        }).catch(() => {          this.msgError('接诊失败')          this.loading = false        })      },      // 载入      handleLoading(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 根据患者ID查询患者信息档案信息 病历信息        getPatientAllMessageByPatientId(patientId).then(res => {          this.patientAllObj.patientObj = res.data.patient          this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)          this.patientAllObj.patientFileObj = res.data.patientFile          this.patientAllObj.careHistoryObjList = res.data.careHistoryList          this.openRegistration = false          this.loading = false          // 查询当前挂号单之前对应的病例信息        }).catch(() => {          this.msgError('载入失败')          this.loading = false        })        // 根据挂号单ID查询对应的病历信息        this.getCareHistoryByRegId(row.regId)      },      // 根据挂号单ID查询对应的病历信息      getCareHistoryByRegId(regId) {        this.loading = true        getCareHistoryByRegId(regId).then(res => {          if (res.data != null) {            this.careHistory = res.data            this.caseDateObj = res.data.careDate            // 如果有数据根据病历ID查询处方信息            this.getCareOrdersByChId(res.data.chId)          } else {            this.resetCareHistoryAndCareOrders(regId)          }          this.loading = false        }).catch(() => {          this.msgError('查询病历失败')          this.loading = false        })      },      // 根据病历ID查询处方信息      getCareOrdersByChId(chId) {        this.loading = true        queryCareOrdersByChId(chId).then(res => {          this.careOrders = res.data          this.loading = false        }).catch(() => {          this.msgError('查询详情失败')          this.loading = false        })      },      // 重置右边的病历和处方详情数据      resetCareHistoryAndCareOrders(regId) {        this.careHistory = {          // 当前就诊中的挂号单ID          regId: regId,          chId: undefined,          caseDate: undefined,          caseDateObj: new Date(),          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        }        this.careOrders = []      },      // 保存病历      handelSaveCareHistory() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        // 封装数据        this.careHistory.caseDate = this.moment(this.caseDateObj).format('YYYY-MM-DD')        this.careHistory.patientId = this.patientAllObj.patientObj.patientId        this.careHistory.patientName = this.patientAllObj.patientObj.name        this.loading = true        saveCareHistory(this.careHistory).then(res => {          this.msgSuccess('保存病历成功')          this.loading = false          this.careHistory.chId = res.data.chId        }).catch(() => {          this.msgError('保存病历失败')          this.loading = false        })      },      // 打开添加药用处方的弹出层      handelAddMedicinesOrder() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        if (!this.careHistory.chId) {          this.msgError('请先保存病历')          return        }        this.submitCareOrder.careOrder.coType = '0'        this.title = '添加【药用】处方'        this.openAddOrderItem = true        this.submitCareOrder.careOrderItems = []      },      // 打开添加检查处方的弹出层      handelAddCheckItemOrder() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        if (!this.careHistory.chId) {          this.msgError('请先保存病历')          return        }        this.submitCareOrder.careOrder.coType = '1'        this.title = '添加【检查】处方'        this.openAddOrderItem = true        this.submitCareOrder.careOrderItems = []      },      // 打开药品或者检查项目的抽屉      handleOpenAddOrderItemDrawer() {        if (this.submitCareOrder.careOrder.coType === '0') {          // 打开药口列表抽屉          this.openDrawerMedicines = true          this.resetItemFormQuery()          this.getMedicinesList()        } else {          // 打开检查项目的抽屉          this.openDrawerCheckItem = true          this.resetItemFormQuery()          this.getCheckItemList()        }      },      // 加载药品数据      getMedicinesList() {        this.tableItemList = []        this.drawerLoading = true        listMedicinesForPage(this.queryItemFormParams).then(res => {          this.tableItemList = res.data          this.total = res.total          this.drawerLoading = false        }).catch(() => {          this.drawerLoading = false          this.msgError('查询药品失败')        })      },      // 数据表格的多选择框选择时触发      handleMedicinesSelectionChange(selection) {        this.selectItemList = selection      },      // 分页pageSize变化时触发      handleMedicinesSizeChange(val) {        this.queryItemFormParams.pageSize = val        // 重新查询        this.getMedicinesList()      },      // 点击上一页  下一页，跳转到哪一页面时触发      handleMedicinesCurrentChange(val) {        this.queryItemFormParams.pageNum = val        // 重新查询        this.getMedicinesList()      },      // 搜索药品的方法      handleMedicinesFormQuery() {        this.queryItemFormParams.pageNum = 1        this.getMedicinesList()      },      // 加载药品数据      getCheckItemList() {        this.tableItemList = []        this.drawerLoading = true        listCheckItemForPage(this.queryItemFormParams).then(res => {          this.tableItemList = res.data          this.total = res.total          this.drawerLoading = false        }).catch(() => {          this.drawerLoading = false          this.msgError('查询检查项目失败')        })      },      // 数据表格的多选择框选择时触发      handleCheckItemSelectionChange(selection) {        this.selectItemList = selection      },      // 分页pageSize变化时触发      handleCheckItemSizeChange(val) {        this.queryItemFormParams.pageSize = val        // 重新查询        this.getCheckItemList()      },      // 点击上一页  下一页，跳转到哪一页面时触发      handleCheckItemCurrentChange(val) {        this.queryItemFormParams.pageNum = val        // 重新查询        this.getCheckItemList()      },      // 搜索检查项目的方法      handleCheckItemFormQuery() {        this.queryItemFormParams.pageNum = 1        this.getCheckItemList()      },      // 重置抽屉的查询条件      resetItemFormQuery() {        this.queryItemFormParams = {          pageNum: 1,          pageSize: 10,          keywords: undefined        }        if (this.submitCareOrder.careOrder.coType === '0') {          this.getMedicinesList()        } else {          this.getCheckItemList()        }      },      // 点击药品或检查项目抽屉页面的添加并关闭按钮      hanldeAddCareItem() {        const coType = this.submitCareOrder.careOrder.coType        if (this.selectItemList.length === 0) {          this.msgError('请选择【' + (coType === '0' ? '药品' : '检查项目') + '】')          return        }        if (coType === '0') { // 药品          this.selectItemList.filter(item => {            const obj = {              itemRefId: item.medicinesId,              itemName: item.medicinesName,              itemType: coType,              num: 1,              price: item.prescriptionPrice,              amount: 1 \* item.prescriptionPrice,              remark: '请按说明服用'            }            let flag = false// 默认里面没有加            this.submitCareOrder.careOrderItems.filter(i => {              if (i.itemRefId === obj.itemRefId) {                i.num = i.num + 1                flag = true// 说明之前加过              }            })            if (!flag) {              this.submitCareOrder.careOrderItems.push(obj)            }            this.openDrawerMedicines = false          })        } else { // 检查项目          this.selectItemList.filter(item => {            const obj = {              itemRefId: item.checkItemId,              itemName: item.checkItemName,              itemType: coType,              num: 1,              price: item.unitPrice,              amount: 1 \* item.unitPrice,              remark: '按要求检查'            }            let flag = false// 默认里面没有加            this.submitCareOrder.careOrderItems.filter(i => {              if (i.itemRefId === obj.itemRefId) {                i.num = i.num + 1                flag = true// 说明之前加过              }            })            if (!flag) {              this.submitCareOrder.careOrderItems.push(obj)            }            this.openDrawerCheckItem = false          })        }        // 计算总价        this.computeOrderItemAllAmount()      },      // 删除弹出层里面的详情      handleCareOrderItemDelete(row) {        this.submitCareOrder.careOrderItems.splice(row.index, 1)        this.computeOrderItemAllAmount()      },      // 把弹出层的表格的数据加上index      tableRowClassName({ row, rowIndex }) {        row.index = rowIndex      },      // 计算当前处方详情的总价      computeOrderItemAllAmount() {        this.submitCareOrder.careOrder.allAmount = 0.00        this.submitCareOrder.careOrderItems.filter(item => {          this.submitCareOrder.careOrder.allAmount +=          (item.num \* item.price)        })      },      // 监听药品或检查项目弹出层的数量的变化      handleCareOrderItemNumChange(row) {        row.amount = row.num \* row.price        this.computeOrderItemAllAmount()      },      // 保存处方单信息      handleSaveOrderItem() {        this.submitCareOrder.careOrder.patientId = this.careHistory.patientId        this.submitCareOrder.careOrder.patientName = this.careHistory.patientName        this.submitCareOrder.careOrder.chId = this.careHistory.chId        console.log(this.submitCareOrder)      }    }  }  </script>  <style scoped>    .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }    .item{      font-size: 14px;      margin-bottom: 8px;    }  </style> |

## 修改src/api/doctor/care.js

|  |
| --- |
| // 保存处方及详情信息  export function saveCareOrderItem(data) {    return request({      url: '/doctor/care/saveCareOrderItem',      method: 'post',      data: data    })  }  // 根据处方详情ID删除处方详情【只能删除未支付的】  export function deleteCareOrderItemById(itemId) {    return request({      url: '/doctor/care/deleteCareOrderItemById/' + itemId,      method: 'delete'    })  }  // 根据挂号单号完成就诊  export function visitComplete(regId) {    return request({      url: '/doctor/care/visitComplete/' + regId,      method: 'post'    })  } |

## 修改src/views/doctor/care/index.vue后端接口联调

|  |
| --- |
| <template>    <div v-loading="loading" class="app-container">      <!-- 选择门诊和急诊的选项开始 -->      <el-card style="margin-bottom:5px;text-align:center">        <el-radio-group v-model="schedulingType" @change="schedulingTypeChange">          <el-radio-button            v-for="dict in schedulingTypeOptions"            :key="dict.dictValue"            :label="dict.dictValue"            :value="dict.dictValue"          >            {{ dict.dictLabel }}          </el-radio-button>        </el-radio-group>      </el-card>      <!-- 选择门诊和急诊的选项结束 -->      <!-- 下面的整体卡片开始 -->      <el-row :gutter="5">        <el-col :span="7">          <!-- 左边的患者信息开始 -->          <el-card>            <el-form ref="form" :model="patientAllObj.patientObj" :inline="true" label-width="88px">              <el-form-item label="患者姓名" prop="name">                <el-input                  v-model="patientAllObj.patientObj.name"                  placeholder="请输入患者姓名"                  :disabled="true"                  style="width:100%"                  size="small"                >                  <el-button slot="append" icon="el-icon-user-solid" @click="viewRegistration" />                </el-input>              </el-form-item>              <el-form-item label="身份证号" prop="idCard">                <el-input                  v-model="patientAllObj.patientObj.idCard"                  placeholder="请输入身份证号"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者电话" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.phone"                  placeholder="请输入患者电话"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="性别" prop="sex">                <el-radio-group v-model="patientAllObj.patientObj.sex">                  <el-radio                    v-for="d in sexOptions"                    :key="d.dictValue"                    :label="d.dictValue"                    :value="d.dictValue"                  >{{ d.dictLabel }}</el-radio>                </el-radio-group>              </el-form-item>              <el-form-item label="出生日期" prop="phone">                <el-input                  v-model="patientAllObj.patientObj.birthday"                  placeholder="请选择出生日期"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者年龄" prop="age">                <el-input                  v-model="patientAllObj.patientObj.age"                  placeholder="请输入患者年龄"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="患者地址" prop="address">                <el-input                  v-model="patientAllObj.patientObj.address"                  placeholder="请输入患者地址"                  :disabled="true"                  style="width:100%"                  size="small"                />              </el-form-item>              <el-form-item label="过敏信息" prop="allergyInfo">                <el-input                  v-model="patientAllObj.patientObj.allergyInfo"                  placeholder="请输入过敏信息"                  type="textarea"                  :autosize="{minRows:2,maxRows:4}"                  :disabled="true"                  maxlength="100"                  show-word-limit                  style="width:100%"                  size="small"                />              </el-form-item>            </el-form>            <el-tabs :stretch="true" type="card">              <el-tab-pane label="患者档案">                <div v-if="patientAllObj.patientObj.isFinal==='1'">                  <div class="item">                    紧急联系人:<span v-text="patientAllObj.patientFileObj.emergencyContactName" />                  </div>                  <div class="item">                    紧急联系人电话:<span v-text="patientAllObj.patientFileObj.emergencyContactPhone" />                  </div>                  <div class="item">                    关系: <span v-text="patientAllObj.patientFileObj.emergencyContactRelation" />                  </div>                  <div class="item">                    左耳听力:<span v-text="patientAllObj.patientFileObj.leftEarHearing" />                  </div>                  <div class="item">                    右耳听力:<span v-text="patientAllObj.patientFileObj.rightEarHearing" />                  </div>                  <div class="item">                    左眼视力:<span v-text="patientAllObj.patientFileObj.leftVision" />                  </div>                  <div class="item">                    右眼视力:<span v-text="patientAllObj.patientFileObj.rightVision" />                  </div>                  <div class="item">                    身高:<span v-text="patientAllObj.patientFileObj.height" />                  </div>                  <div class="item">                    体重:<span v-text="patientAllObj.patientFileObj.weight" />                  </div>                  <div class="item">                    血型:<span v-text="patientAllObj.patientFileObj.bloodType" />                  </div>                  <div class="item">                    个人史:<span v-text="patientAllObj.patientFileObj.personalInfo" />                  </div>                  <div class="item">                    家族史:<span v-text="patientAllObj.patientFileObj.familyInfo" />                  </div>                  <div class="item">                    档案创建时间:<span v-text="patientAllObj.patientFileObj.createTime" />                  </div>                  <div class="item">                    档案更新时间:<span v-text="patientAllObj.patientFileObj.updateTime" />                  </div>                </div>                <div v-else style="text-align:center">                  暂无患者档案信息                </div>              </el-tab-pane>              <el-tab-pane label="历史病历">                <div v-if="patientAllObj.careHistoryObjList.length>0">                  <el-collapse accordion>                    <el-collapse-item v-for="(item,index) in patientAllObj.careHistoryObjList" :key="index">                      <template slot="title">                        就诊部门:【{{ item.deptName }}】 就诊时间:{{ item.careDate }}                      </template>                      <div class="item">                        主诉:{{ item.caseTitle }}                      </div>                      <div class="item">                        发病日期:{{ item.caseDate }}                      </div>                      <div class="item">                        诊断信息:{{ item.caseResult }}                      </div>                      <div class="item">                        医生建议:{{ item.doctorTips }}                      </div>                      <div class="item">                        备注:{{ item.remark }}                      </div>                      <div class="item">                        接诊类型:{{ item.receiveType==='0'?'初诊':'复诊' }}                      </div>                      <div class="item">                        是否传染:{{ item.isContagious==='0'?'否':'是' }}                      </div>                    </el-collapse-item>                  </el-collapse>                </div>                <div v-else style="text-align:center">                  暂无患者病历信息                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 左边的患者信息结束 -->        </el-col>        <el-col :span="17">          <!-- 右边的病历，处方开始 -->          <el-card>            <el-row :gutter="5">              <el-col :span="12">                挂号单号:                <span v-if="careHistory.regId===undefined" style="color:red">选择患者后显示</span>                <span v-else style="color:red">{{ careHistory.regId }}</span>                <br>                病历编号:                <span v-if="careHistory.chId===undefined" style="color:red">保存病例后显示</span>                <span v-else style="color:blue">{{ careHistory.chId }}</span>              </el-col>              <el-col :span="12" style="text-align:right">                <el-button type="primary" :disabled="careHistory.regId===undefined" icon="el-icon-check" @click="handleSaveCareHistory">保存病历</el-button>                <el-button type="danger" :disabled="careHistory.regId===undefined" icon="el-icon-finished" @click="handleVisitComplete">完成就诊</el-button>              </el-col>            </el-row>          </el-card>          <el-card style="margin-top:3px">            <el-tabs :v-model="careActiveName" type="card">              <el-tab-pane label="病历">                <!-- 病例表单开始 -->                <el-card>                  <el-form ref="form" :model="careHistory" :inline="true" label-width="88px">                    <el-form-item label="发病日期" prop="caseDateObj">                      <el-date-picker                        v-model="caseDateObj"                        value-format="yyyy-MM-dd"                        size="small"                      />                    </el-form-item>                    <el-form-item label="接诊类型" prop="receiveType">                      <el-select                        v-model="careHistory.receiveType"                        placeholder="接诊类型"                        size="small"                      >                        <el-option                          v-for="dict in receiveTypeOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                    <el-form-item label="是否传染" prop="isContagious">                      <el-select                        v-model="careHistory.isContagious"                        placeholder="是否传染"                        size="small"                      >                        <el-option                          v-for="dict in isContagiousOptions"                          :key="dict.dictValue"                          :label="dict.dictLabel"                          :value="dict.dictValue"                        />                      </el-select>                    </el-form-item>                  </el-form>                  <el-form ref="form" :model="careHistory" label-width="88px">                    <el-form-item label="主诉" :inline="false" prop="caseTitle">                      <el-input                        v-model="careHistory.caseTitle"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入主诉"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="诊断信息" :inline="false" prop="caseResult">                      <el-input                        v-model="careHistory.caseResult"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入诊断信息"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="医生建议" :inline="false" prop="doctorTips">                      <el-input                        v-model="careHistory.doctorTips"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入医生建议"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                    <el-form-item label="备注" :inline="false" prop="remark">                      <el-input                        v-model="careHistory.remark"                        type="textarea"                        :autosize="{ minRows: 4, maxRows: 6}"                        placeholder="请输入备注"                        style="width:100%"                        maxlength="200"                        show-word-limit                        size="small"                      />                    </el-form-item>                  </el-form>                </el-card>                <!-- 病例表单结束 -->              </el-tab-pane>              <el-tab-pane label="处方">                <!-- 处方详情开始 -->                <el-collapse v-if="careOrders.length>0" accordion>                  <el-collapse-item v-for="(item,index) in careOrders" :key="index">                    <template slot="title">                      【{{ item.careOrder.coType==='0'?'药用处方':'检查处方' }}】【{{ index+1 }}】【处方总额】:￥{{ item.careOrder.allAmount }}                    </template>                    <el-table                      v-loading="loading"                      border                      :data="item.careOrderItems"                    >                      <el-table-column label="序号" align="center" type="index" width="50" />                      <el-table-column :label="item.careOrder.coType==='0'?'药品名称':'项目名称'" align="center" prop="itemName" />                      <el-table-column label="数量" align="center" prop="num" />                      <el-table-column label="单价(元)" align="center" prop="price" />                      <el-table-column label="金额(元)" align="center" prop="amount" />                      <el-table-column label="备注" align="center" prop="remark" />                      <el-table-column label="状态" align="center" prop="status" :formatter="orderDetailsStatusFormatter" />                      <el-table-column label="操作" align="center">                        <template slot-scope="scope">                          <el-button v-show="scope.row.status==='0'" type="text" icon="el-icon-delete" size="mini" @click="handleCareOrderItemDeleteByItemId(scope.row)">删除</el-button>                        </template>                      </el-table-column>                    </el-table>                  </el-collapse-item>                </el-collapse>                <!-- 处方详情结束 -->                <div style="margin:3px;text-align:left">                  <el-button type="success" icon="el-icon-plus" :disabled="careHistory.chId===undefined" @click="handelAddMedicinesOrder">添加药用处方</el-button>                  <el-button type="success" icon="el-icon-plus" :disabled="careHistory.chId===undefined" @click="handelAddCheckItemOrder">添加检查处方</el-button>                </div>              </el-tab-pane>            </el-tabs>          </el-card>          <!-- 右边的病历，处方结束 -->        </el-col>      </el-row>      <!-- 下面的整体卡片结束 -->      <!-- 弹出选择患者的层开始 -->      <el-dialog        title="请选择挂号患者"        :visible.sync="openRegistration"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <el-tabs v-model="activeName" :stretch="true" type="card" @tab-click="handleRegistrationTabClick">          <el-tab-pane label="待就诊列表" name="t1">            <el-table v-loading="tableLoading" border :data="toBeSeenRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleVisit(scope.row)">接诊</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊中列表" name="t2">            <el-table v-loading="tableLoading" border :data="visitingRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>          <el-tab-pane label="就诊完成列表" name="t3">            <el-table v-loading="tableLoading" border :data="visitCompletedRegistration">              <el-table-column type="expand">                <template slot-scope="scope">                  <el-form label-position="right" inline class="demo-table-expand">                    <el-form-item label="挂号ID">                      <span>{{ scope.row.regId }}</span>                    </el-form-item>                    <el-form-item label="患者ID">                      <span>{{ scope.row.patientId }}</span>                    </el-form-item>                    <el-form-item label="挂号员">                      <span>{{ scope.row.createBy }}</span>                    </el-form-item>                    <el-form-item label="挂号时间">                      <span>{{ scope.row.createTime }}</span>                    </el-form-item>                  </el-form>                </template>              </el-table-column>              <el-table-column label="患者姓名" align="center" prop="patientName" />              <el-table-column label="流水编号" align="center" prop="regNumber" />              <el-table-column label="挂号类型" align="center" prop="schedulingType" :formatter="schedulingTypeFormatter" />              <el-table-column label="操作" align="center">                <template slot-scope="scope">                  <el-button type="success" icon="el-icon-check" size="mini" @click="handleLoading(scope.row)">载入</el-button>                </template>              </el-table-column>            </el-table>          </el-tab-pane>        </el-tabs>      </el-dialog>      <!-- 弹出选择患者的层结束 -->      <!-- 添加药用处方和检查出方的弹出层开始 -->      <el-dialog        :title="title"        :visible.sync="openAddOrderItem"        width="1000px"        center        :close-on-click-modal="false"        append-to-body      >        <div style="margin:3px;text-align:right">          【{{ submitCareOrder.careOrder.coType==='0'?'药用':'检查' }}】处方总额:￥<span style="color:red">{{ submitCareOrder.careOrder.allAmount }}</span>          <el-button style="margin-left:15px" type="success" icon="el-icon-plus" :disabled="submitCareOrder.careOrderItems.length===0" @click="handleSaveOrderItem">确定添加</el-button>        </div>        <el-table          border          :data="submitCareOrder.careOrderItems"          :row-class-name="tableRowClassName"        >          <el-table-column label="序号" align="center" type="index" width="50" />          <el-table-column :label="submitCareOrder.careOrder.coType==='0'?'药品名称':'项目名称'" align="center" prop="itemName" />          <el-table-column label="数量" width="180px" align="center" prop="num">            <template slot-scope="scope">              <el-input-number                v-model="scope.row.num"                size="small"                @change="handleCareOrderItemNumChange(scope.row)"              />            </template>          </el-table-column>          <el-table-column label="单价(元)" align="center" prop="price" />          <el-table-column label="金额(元)" align="center" prop="amount" />          <el-table-column label="备注" align="center" prop="remark">            <template slot-scope="scope">              <el-input                v-model="scope.row.remark"                size="small"              />            </template>          </el-table-column>          <el-table-column label="操作" align="center">            <template slot-scope="scope">              <el-button type="text" icon="el-icon-delete" size="mini" @click="handleCareOrderItemDelete(scope.row)">删除</el-button>            </template>          </el-table-column>        </el-table>        <el-button type="primary" size="mini" style="width:100%;margin-top:2px" @click="handleOpenAddOrderItemDrawer">添加【{{ submitCareOrder.careOrder.coType==='0'?'药品':'检查' }}】项 </el-button>      </el-dialog>      <!-- 添加药用处方和检查处方的弹出层结束 -->      <!-- 药品列表的抽屉开始 -->      <el-drawer        :visible.sync="openDrawerMedicines"        direction="rtl"        append-to-body      >        <h2 align="center">药品列表</h2>        <el-form ref="queryItemForm" :model="queryItemFormParams" label-width="68px">          <el-form-item label="关键字" prop="keywords">            <el-input              v-model="queryItemFormParams.keywords"              placeholder="请输入关键字"              clearable              size="small"              style="width:180px"            />            <el-button type="primary" icon="el-icon-search" size="mini" @click="handleMedicinesFormQuery">查询</el-button>            <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetItemFormQuery">重置</el-button>          </el-form-item>          <el-table            v-loading="drawerLoading"            height="600px"            border            :data="tableItemList"            @selection-change="handleMedicinesSelectionChange"          >            <el-table-column type="selection" width="55" align="center" />            <el-table-column label="药品ID" align="center" prop="medicinesId" />            <el-table-column label="药品名称" align="center" prop="medicinesName" />            <el-table-column label="关键字" align="center" prop="keywords" />            <el-table-column label="库存量" align="center" prop="medicinesStockNum" />            <el-table-column label="单位" align="center">              <template slot-scope="scope">                {{ scope.row.conversion }}/ {{ scope.row.unit }}              </template>            </el-table-column>          </el-table>          <!-- 数据表格结束 -->          <!-- 分页控件开始 -->          <el-pagination            v-show="total>0"            :current-page="queryItemFormParams.pageNum"            :page-sizes="[5, 10, 20, 30]"            :page-size="queryItemFormParams.pageSize"            layout="total, sizes, prev, pager, next, jumper"            :total="total"            @size-change="handleMedicinesSizeChange"            @current-change="handleMedicinesCurrentChange"          />          <el-form-item>            <div style="text-align:center">              <el-button type="primary" icon="el-icon-check" size="mini" @click="hanldeAddCareItem">添加并关闭</el-button>            </div>          </el-form-item>        </el-form>      <!-- 分页控件结束 -->      </el-drawer>      <!-- 药品列表的抽屉结束 -->      <!-- 检查项目的抽屉开始 -->      <el-drawer        :visible.sync="openDrawerCheckItem"        direction="rtl"        append-to-body      >        <h2 align="center">检查项目列表</h2>        <el-form ref="queryItemForm" :model="queryItemFormParams" label-width="68px">          <el-form-item label="关键字" prop="keywords">            <el-input              v-model="queryItemFormParams.keywords"              placeholder="请输入关键字"              clearable              size="small"              style="width:180px"            />            <el-button type="primary" icon="el-icon-search" size="mini" @click="handleCheckItemFormQuery">查询</el-button>            <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetItemFormQuery">重置</el-button>          </el-form-item>          <el-table            v-loading="drawerLoading"            height="600px"            border            :data="tableItemList"            @selection-change="handleCheckItemSelectionChange"          >            <el-table-column type="selection" width="55" align="center" />            <el-table-column label="项目ID" align="center" prop="checkItemId" />            <el-table-column label="项目名称" align="center" prop="checkItemName" />            <el-table-column label="关键字" align="center" prop="keywords" />            <el-table-column label="项目单价" align="center" prop="unitPrice" />            <el-table-column label="单位" align="center" prop="unit" />          </el-table>          <!-- 数据表格结束 -->          <!-- 分页控件开始 -->          <el-pagination            v-show="total>0"            :current-page="queryItemFormParams.pageNum"            :page-sizes="[5, 10, 20, 30]"            :page-size="queryItemFormParams.pageSize"            layout="total, sizes, prev, pager, next, jumper"            :total="total"            @size-change="handleCheckItemSizeChange"            @current-change="handleCheckItemCurrentChange"          />          <el-form-item>            <div style="text-align:center">              <el-button type="primary" icon="el-icon-check" size="mini" @click="hanldeAddCareItem">添加并关闭</el-button>            </div>          </el-form-item>        </el-form>      </el-drawer>      <!-- 检查项目的抽屉结束 -->    </div>  </template>  <script>  import { queryToBeSeenRegistration, queryVisitingRegistration, queryVisitCompletedRegistration,    receivePatient, getPatientAllMessageByPatientId,    saveCareHistory, getCareHistoryByRegId, queryCareOrdersByChId,    saveCareOrderItem, deleteCareOrderItemById, visitComplete  } from '@/api/docter/care'  import { listCheckItemForPage } from '@/api/system/checkItem'  import { listMedicinesForPage } from '@/api/erp/medicines'  export default {    data() {      return {        // 总的遮罩层        loading: false,        // 表格的遮罩层        tableLoading: false,        // 门诊急诊的默认值        schedulingType: '1',        // 就诊类型的字典数据        schedulingTypeOptions: [],        // 排班类型的字典数据        subsectionTypeOptions: [],        // 性别数据字典        sexOptions: [],        // 接诊类型字典        receiveTypeOptions: [],        // 是否传染字典        isContagiousOptions: [],        // 处方详情的状态字典数据        orderDetailsStatusOptions: [],        // 是否打开选择挂号患者的弹层        openRegistration: false,        // 当前选中的挂号选项卡的        activeName: 't1',        // 右边的病例和处方的选项卡        careActiveName: 'c1',        // 待就诊数据        toBeSeenRegistration: [],        // 就诊中的数据        visitingRegistration: [],        // 就诊完成的数据        visitCompletedRegistration: [],        // 发病日期        caseDateObj: new Date(),        // 左边患者数据        patientAllObj: {          patientObj: {},          patientFileObj: {},          careHistoryObjList: []        },        // 提交到数据库的对象        careHistory: {          // 当前就诊中的挂号单ID          regId: undefined,          chId: undefined,          caseDate: undefined,          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        },        // 存放处方及详情数据        careOrders: [],        // 最后要提交到的处方及详情        submitCareOrder: {          careOrder: {            allAmount: 0.00,            patientId: undefined,            patientName: undefined,            coType: '0' // 默认为药用处方          },          careOrderItems: []        },        // 弹出层的标题        title: '',        // 是否打开药品和检查项的添加的弹出层        openAddOrderItem: false,        // 是否打开药品列表的抽屉        openDrawerMedicines: false,        // 是否打开检查项目列表的抽屉        openDrawerCheckItem: false,        // 药品和检查项目抽屉的查询条件        queryItemFormParams: {          pageNum: 1,          pageSize: 10,          keywords: undefined        },        // 药品或检查项目表格的数据        tableItemList: [],        // 抽屉里面数据的总条数        total: 0,        // 抽屉数据加载的遮罩        drawerLoading: false,        // 抽屉里面选中的数据        selectItemList: []      }    },    created() {      // 加载排班类型 1 门诊  2急诊      this.getDataByType('his\_scheduling\_type').then(res => {        this.schedulingTypeOptions = res.data      })      // 排班时段 1 2 3 上午 下午 晚上      this.getDataByType('his\_subsection\_type').then(res => {        this.subsectionTypeOptions = res.data      })      // 加载性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })      //  加载接诊类型字典      this.getDataByType('his\_receive\_type').then(res => {        this.receiveTypeOptions = res.data      })      //  加载是否传染字典      this.getDataByType('his\_contagious\_status').then(res => {        this.isContagiousOptions = res.data      })      //  加载处方详情的状态字典数据      this.getDataByType('his\_order\_details\_status').then(res => {        this.orderDetailsStatusOptions = res.data      })    },    methods: {      // 门诊急诊切换事件      schedulingTypeChange(value) {        this.schedulingType = value      },      // 打开选择挂号患者的弹出层      viewRegistration() {        this.activeName = 't1'        this.openRegistration = true        this.queryToBeSeenRegistration()      },      // 挂号患者弹出层的选项卡change事件      handleRegistrationTabClick(tab, event) {        if (tab.name === 't1') {          this.queryToBeSeenRegistration()        } else if (tab.name === 't2') {          this.queryVisitingRegistration()        } else if (tab.name === 't3') {          this.queryVisitCompletedRegistration()        }      },      // 查询待就诊的数据      queryToBeSeenRegistration() {        this.tableLoading = true        queryToBeSeenRegistration(this.schedulingType).then(res => {          this.toBeSeenRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊中的数据      queryVisitingRegistration() {        this.tableLoading = true        queryVisitingRegistration(this.schedulingType).then(res => {          this.visitingRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 查询就诊完成的数据      queryVisitCompletedRegistration() {        this.tableLoading = true        queryVisitCompletedRegistration(this.schedulingType).then(res => {          this.visitCompletedRegistration = res.data          this.tableLoading = false        }).catch(() => {          this.msgError('查询失败')          this.tableLoading = false        })      },      // 翻译挂号类型      schedulingTypeFormatter(row) {        return this.selectDictLabel(this.schedulingTypeOptions, row.schedulingType)      },      // 翻译处方详情状态      orderDetailsStatusFormatter(row) {        return this.selectDictLabel(this.orderDetailsStatusOptions, row.status)      },      // 接诊      handleVisit(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 接诊，更新挂号的状态为就诊中并写入接诊的医生信息        this.loading = true        receivePatient(this.careHistory.regId).then(res => {          this.loading = false          this.msgSuccess('接诊成功')          // 根据患者ID查询患者信息档案信息 病历信息          getPatientAllMessageByPatientId(patientId).then(res => {            this.patientAllObj.patientObj = res.data.patient            this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)            this.patientAllObj.patientFileObj = res.data.patientFile            this.patientAllObj.careHistoryObjList = res.data.careHistoryList            this.openRegistration = false            this.loading = false          }).catch(() => {            this.msgError('查询患者信息失败')            this.loading = false          })        }).catch(() => {          this.msgError('接诊失败')          this.loading = false        })      },      // 载入      handleLoading(row) {        this.careHistory.regId = row.regId        const patientId = row.patientId        // 根据患者ID查询患者信息档案信息 病历信息        getPatientAllMessageByPatientId(patientId).then(res => {          this.patientAllObj.patientObj = res.data.patient          this.patientAllObj.patientObj.age = this.getAge(res.data.patient.birthday)          this.patientAllObj.patientFileObj = res.data.patientFile          this.patientAllObj.careHistoryObjList = res.data.careHistoryList          this.openRegistration = false          this.loading = false          // 查询当前挂号单之前对应的病例信息        }).catch(() => {          this.msgError('载入失败')          this.loading = false        })        // 根据挂号单ID查询对应的病历信息        this.getCareHistoryByRegId(row.regId)      },      // 根据挂号单ID查询对应的病历信息      getCareHistoryByRegId(regId) {        this.loading = true        getCareHistoryByRegId(regId).then(res => {          if (res.data != null) {            this.careHistory = res.data            this.caseDateObj = res.data.careDate            // 如果有数据根据病历ID查询处方信息            this.getCareOrdersByChId(res.data.chId)          } else {            this.resetCareHistoryAndCareOrders(regId)          }          this.loading = false        }).catch(() => {          this.msgError('查询病历失败')          this.loading = false        })      },      // 根据病历ID查询处方信息      getCareOrdersByChId(chId) {        this.loading = true        queryCareOrdersByChId(chId).then(res => {          this.careOrders = res.data          this.loading = false        }).catch(() => {          this.msgError('查询详情失败')          this.loading = false        })      },      // 重置右边的病历和处方详情数据      resetCareHistoryAndCareOrders(regId) {        this.careHistory = {          // 当前就诊中的挂号单ID          regId: regId,          chId: undefined,          caseDate: undefined,          caseDateObj: new Date(),          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        }        this.careOrders = []      },      // 保存病历      handleSaveCareHistory() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        // 封装数据        this.careHistory.caseDate = this.moment(this.caseDateObj).format('YYYY-MM-DD')        this.careHistory.patientId = this.patientAllObj.patientObj.patientId        this.careHistory.patientName = this.patientAllObj.patientObj.name        this.loading = true        saveCareHistory(this.careHistory).then(res => {          this.msgSuccess('保存病历成功')          this.loading = false          this.careHistory.chId = res.data.chId        }).catch(() => {          this.msgError('保存病历失败')          this.loading = false        })      },      // 打开添加药用处方的弹出层      handelAddMedicinesOrder() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        if (!this.careHistory.chId) {          this.msgError('请先保存病历')          return        }        this.submitCareOrder.careOrder.coType = '0'        this.title = '添加【药用】处方'        this.openAddOrderItem = true        this.submitCareOrder.careOrderItems = []      },      // 打开添加检查处方的弹出层      handelAddCheckItemOrder() {        if (!this.careHistory.regId) {          this.msgError('请选择挂号患者')          return        }        if (!this.careHistory.chId) {          this.msgError('请先保存病历')          return        }        this.submitCareOrder.careOrder.coType = '1'        this.title = '添加【检查】处方'        this.openAddOrderItem = true        this.submitCareOrder.careOrderItems = []      },      // 打开药品或者检查项目的抽屉      handleOpenAddOrderItemDrawer() {        if (this.submitCareOrder.careOrder.coType === '0') {          // 打开药口列表抽屉          this.openDrawerMedicines = true          this.resetItemFormQuery()          this.getMedicinesList()        } else {          // 打开检查项目的抽屉          this.openDrawerCheckItem = true          this.resetItemFormQuery()          this.getCheckItemList()        }      },      // 加载药品数据      getMedicinesList() {        this.tableItemList = []        this.drawerLoading = true        listMedicinesForPage(this.queryItemFormParams).then(res => {          this.tableItemList = res.data          this.total = res.total          this.drawerLoading = false        }).catch(() => {          this.drawerLoading = false          this.msgError('查询药品失败')        })      },      // 数据表格的多选择框选择时触发      handleMedicinesSelectionChange(selection) {        this.selectItemList = selection      },      // 分页pageSize变化时触发      handleMedicinesSizeChange(val) {        this.queryItemFormParams.pageSize = val        // 重新查询        this.getMedicinesList()      },      // 点击上一页  下一页，跳转到哪一页面时触发      handleMedicinesCurrentChange(val) {        this.queryItemFormParams.pageNum = val        // 重新查询        this.getMedicinesList()      },      // 搜索药品的方法      handleMedicinesFormQuery() {        this.queryItemFormParams.pageNum = 1        this.getMedicinesList()      },      // 加载药品数据      getCheckItemList() {        this.tableItemList = []        this.drawerLoading = true        listCheckItemForPage(this.queryItemFormParams).then(res => {          this.tableItemList = res.data          this.total = res.total          this.drawerLoading = false        }).catch(() => {          this.drawerLoading = false          this.msgError('查询检查项目失败')        })      },      // 数据表格的多选择框选择时触发      handleCheckItemSelectionChange(selection) {        this.selectItemList = selection      },      // 分页pageSize变化时触发      handleCheckItemSizeChange(val) {        this.queryItemFormParams.pageSize = val        // 重新查询        this.getCheckItemList()      },      // 点击上一页  下一页，跳转到哪一页面时触发      handleCheckItemCurrentChange(val) {        this.queryItemFormParams.pageNum = val        // 重新查询        this.getCheckItemList()      },      // 搜索检查项目的方法      handleCheckItemFormQuery() {        this.queryItemFormParams.pageNum = 1        this.getCheckItemList()      },      // 重置抽屉的查询条件      resetItemFormQuery() {        this.queryItemFormParams = {          pageNum: 1,          pageSize: 10,          keywords: undefined        }        if (this.submitCareOrder.careOrder.coType === '0') {          this.getMedicinesList()        } else {          this.getCheckItemList()        }      },      // 点击药品或检查项目抽屉页面的添加并关闭按钮      hanldeAddCareItem() {        const coType = this.submitCareOrder.careOrder.coType        if (this.selectItemList.length === 0) {          this.msgError('请选择【' + (coType === '0' ? '药品' : '检查项目') + '】')          return        }        if (coType === '0') { // 药品          this.selectItemList.filter(item => {            const obj = {              itemRefId: item.medicinesId,              itemName: item.medicinesName,              itemType: coType,              num: 1,              price: item.prescriptionPrice,              amount: 1 \* item.prescriptionPrice,              remark: '请按说明服用'            }            let flag = false// 默认里面没有加            this.submitCareOrder.careOrderItems.filter(i => {              if (i.itemRefId === obj.itemRefId) {                i.num = i.num + 1                flag = true// 说明之前加过              }            })            if (!flag) {              this.submitCareOrder.careOrderItems.push(obj)            }            this.openDrawerMedicines = false          })        } else { // 检查项目          this.selectItemList.filter(item => {            const obj = {              itemRefId: item.checkItemId,              itemName: item.checkItemName,              itemType: coType,              num: 1,              price: item.unitPrice,              amount: 1 \* item.unitPrice,              remark: '按要求检查'            }            let flag = false// 默认里面没有加            this.submitCareOrder.careOrderItems.filter(i => {              if (i.itemRefId === obj.itemRefId) {                i.num = i.num + 1                flag = true// 说明之前加过              }            })            if (!flag) {              this.submitCareOrder.careOrderItems.push(obj)            }            this.openDrawerCheckItem = false          })        }        // 计算总价        this.computeOrderItemAllAmount()      },      // 删除弹出层里面的详情      handleCareOrderItemDelete(row) {        this.submitCareOrder.careOrderItems.splice(row.index, 1)        this.computeOrderItemAllAmount()      },      // 把弹出层的表格的数据加上index      tableRowClassName({ row, rowIndex }) {        row.index = rowIndex      },      // 计算当前处方详情的总价      computeOrderItemAllAmount() {        this.submitCareOrder.careOrder.allAmount = 0.00        this.submitCareOrder.careOrderItems.filter(item => {          this.submitCareOrder.careOrder.allAmount +=          (item.num \* item.price)        })      },      // 监听药品或检查项目弹出层的数量的变化      handleCareOrderItemNumChange(row) {        row.amount = row.num \* row.price        this.computeOrderItemAllAmount()      },      // 保存处方单信息      handleSaveOrderItem() {        if (this.submitCareOrder.careOrderItems.length === 0) {          this.msgError('添加处方详情')          return        }        this.submitCareOrder.careOrder.patientId = this.careHistory.patientId        this.submitCareOrder.careOrder.patientName = this.careHistory.patientName        this.submitCareOrder.careOrder.chId = this.careHistory.chId        console.log(this.submitCareOrder)        this.loading = true        saveCareOrderItem(this.submitCareOrder).then(res => {          this.loading = false          this.msgSuccess('保存成功')          // 刷新处方列表          this.getCareOrdersByChId(this.careHistory.chId)          // 关闭当前弹出层          this.openAddOrderItem = false        }).catch(() => {          this.msgError('保存失败')          this.loading = false        })      },      // 根据详情ID删除数据库里面的详情【只能删除未支付的】      handleCareOrderItemDeleteByItemId(row) {        const itemId = row.itemId        const itemName = row.itemName        const tx = this        tx.$confirm('是否确定删除【' + itemName + '】这条详情, 是否继续?', '提示', {          confirmButtonText: '确定',          cancelButtonText: '取消',          type: 'warning'        }).then(() => {          deleteCareOrderItemById(itemId).then(res => {            this.msgSuccess('删除成功')            // 刷新处方列表            this.getCareOrdersByChId(this.careHistory.chId)          }).catch(() => {            this.msgError('删除失败')          })        }).catch(() => {          tx.msgError('删除已取消')          tx.loading = false        })      },      // 完成就诊      handleVisitComplete() {        const regId = this.careHistory.regId        const patientName = this.careHistory.patientName        const tx = this        tx.$confirm('是否确定完成【' + patientName + '】的就诊, 是否继续?', '提示', {          confirmButtonText: '确定',          cancelButtonText: '取消',          type: 'warning'        }).then(() => {          visitComplete(regId).then(res => {            this.msgSuccess('操作成功')            // 重置所有数据            this.resetAllData()          }).catch(() => {            this.msgError('操作失败')          })        }).catch(() => {          tx.msgError('操作已取消')          tx.loading = false        })      },      // 重置所有数据      resetAllData() {        this.patientAllObj = {          patientObj: {},          patientFileObj: {},          careHistoryObjList: []        }        this.careHistory = {          // 当前就诊中的挂号单ID          regId: undefined,          chId: undefined,          caseDate: undefined,          receiveType: '0',          isContagious: '0',          caseTitle: undefined,          caseResult: undefined,          doctorTips: undefined,          remark: undefined,          patientId: undefined,          patientName: undefined        }        this.careOrders = []        this.submitCareOrder = {          careOrder: {            allAmount: 0.00,            patientId: undefined,            patientName: undefined,            coType: '0' // 默认为药用处方          },          careOrderItems: []        }      }    }  }  </script>  <style scoped>    .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }    .item{      font-size: 14px;      margin-bottom: 8px;    }  </style> |

# 患者管理-补充

## 修改src/api/doctor/patient.js

|  |
| --- |
| // 根据患者ID查询患者所有信息【基础，档案，病例】  export function getPatientAllMessageByPatientId(patientId) {    return request({      url: '/doctor/patient/getPatientAllMessageByPatientId/' + patientId,      method: 'get'    })  } |

## 修改src/views/doctor/patient/index.vue

|  |
| --- |
| <template>    <div class="app-container">      <!-- 查询条件开始 -->      <el-form ref="queryForm" :model="queryParams" :inline="true" label-width="98px">        <el-form-item label="患者姓名" prop="name">          <el-input            v-model="queryParams.name"            placeholder="请输入患者姓名"            clearable            size="small"            style="width:180px"          />        </el-form-item>        <el-form-item label="手机号" prop="phone">          <el-input            v-model="queryParams.phone"            placeholder="请输入手机号"            clearable            size="small"            style="width:180px"          />        </el-form-item>        <el-form-item label="身份证号" prop="idCard">          <el-input            v-model="queryParams.idCard"            placeholder="请输身份证号"            clearable            size="small"            style="width:180px"          />        </el-form-item>        <el-form-item>          <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>          <el-button type="primary" icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>        </el-form-item>      </el-form>      <!-- 查询条件结束 -->      <!-- 表格工具按钮开始 -->      <el-row :gutter="10" style="margin-bottom: 8px;">        <el-col :span="1.5">          <el-button type="primary" icon="el-icon-plus" size="mini" :disabled="single" @click="handleViewFile">查看档案</el-button>        </el-col>        <el-col :span="1.5">          <el-button type="success" icon="el-icon-edit" size="mini" :disabled="single" @click="handleViewCareHistory">查看就诊病例</el-button>        </el-col>      </el-row>      <!-- 表格工具按钮结束 -->      <!-- 数据表格开始 -->      <el-table v-loading="loading" border :data="patientTableList" @selection-change="handleSelectionChange">        <el-table-column type="selection" width="55" align="center" />        <el-table-column label="姓名" align="center" prop="name" />        <el-table-column label="电话" align="center" prop="phone" />        <el-table-column label="身份证号" align="center" prop="idCard" />        <el-table-column label="出生年月" align="center" prop="birthday">          <template slot-scope="scope">            <span>{{ scope.row.birthday.substring(0,10) }}</span>          </template>        </el-table-column>        <el-table-column label="年龄" align="center" prop="phone" :formatter="calculateAgeFormatter" />        <el-table-column label="性别" prop="sex" align="center" :formatter="sexFormatter" />        <el-table-column label="信息状态" prop="isFinal" align="center" :formatter="isFinalFormatter" />        <el-table-column label="创建时间" align="center" prop="createTime" />      </el-table>      <!-- 数据表格结束 -->      <!-- 分页控件开始 -->      <el-pagination        v-show="total>0"        :current-page="queryParams.pageNum"        :page-sizes="[5, 10, 20, 30]"        :page-size="queryParams.pageSize"        layout="total, sizes, prev, pager, next, jumper"        :total="total"        @size-change="handleSizeChange"        @current-change="handleCurrentChange"      />      <!-- 分页控件结束 -->      <!-- 查看患者档案弹出层开始 -->      <el-dialog        :title="title"        :visible.sync="fileOpen"        width="1000px"        center        append-to-body      >        <el-form label-position="left" label-width="120px" inline class="demo-table-expand">          <el-form-item label="ID:">            <span v-text="patientObj.patientId" />          </el-form-item>          <el-form-item label="姓名:">            <span v-text="patientObj.name" />          </el-form-item>          <el-form-item label="身份证号:">            <span v-text="patientObj.idCard" />          </el-form-item>          <el-form-item label="电话:">            <span v-text="patientObj.phone" />          </el-form-item>          <el-form-item label="患者性别:">            <span v-text="patientObj.sex" />          </el-form-item>          <el-form-item label="出生年月:">            <span v-text="patientObj.birthday" />          </el-form-item>          <el-form-item label="地址信息:">            <span v-text="patientObj.address" />          </el-form-item>          <el-form-item label="过敏信息:">            <span v-text="patientObj.allergyInfo" />          </el-form-item>          <el-form-item label="是否完善信息:">            <span v-text="patientObj.isFinal" />          </el-form-item>          <el-form-item label="最后登录ip:">            <span v-text="patientObj.lastLoginIp" />          </el-form-item>          <el-form-item label="最后登录时间:">            <span v-text="patientObj.lastLoginTime" />          </el-form-item>          <el-form-item label="创建时间:">            <span v-text="patientObj.createTime" />          </el-form-item>          <el-form-item label="更新时间:">            <span v-text="patientObj.updateTime" />          </el-form-item>          <el-form-item label="紧急联系人:">            <span v-text="patientFileObj.emergencyContactName" />          </el-form-item>          <el-form-item label="紧急联系人电话:">            <span v-text="patientFileObj.emergencyContactPhone" />          </el-form-item>          <el-form-item label="关系:">            <span v-text="patientFileObj.emergencyContactRelation" />          </el-form-item>          <el-form-item label="左耳听力:">            <span v-text="patientFileObj.leftEarHearing" />          </el-form-item>          <el-form-item label="右耳听力:">            <span v-text="patientFileObj.rightEarHearing" />          </el-form-item>          <el-form-item label="左眼视力:">            <span v-text="patientFileObj.leftVision" />          </el-form-item>          <el-form-item label="右眼视力:">            <span v-text="patientFileObj.rightVision" />          </el-form-item>          <el-form-item label="身高:">            <span v-text="patientFileObj.height" />          </el-form-item>          <el-form-item label="体重:">            <span v-text="patientFileObj.weight" />          </el-form-item>          <el-form-item label="血型:">            <span v-text="patientFileObj.bloodType" />          </el-form-item>          <el-form-item label="个人史:">            <span v-text="patientFileObj.personalInfo" />          </el-form-item>          <el-form-item label="家族史:">            <span v-text="patientFileObj.familyInfo" />          </el-form-item>          <el-form-item label="档案创建时间:">            <span v-text="patientFileObj.createTime" />          </el-form-item>          <el-form-item label="档案更新时间:">            <span v-text="patientFileObj.updateTime" />          </el-form-item>        </el-form>        <span slot="footer" class="dialog-footer">          <el-button @click="cancelFile">取 消</el-button>        </span>      </el-dialog>      <!-- 查看患者档案弹出层结束 -->      <!-- 病例信息弹出层开始 -->      <el-dialog        :title="title"        :visible.sync="careHistoryOpen"        width="1000px"        center        append-to-body      >        <el-form label-position="left" label-width="120px" inline class="demo-table-expand">          <el-card>            <el-form-item label="ID:">              <span v-text="patientObj.patientId" />            </el-form-item>            <el-form-item label="姓名:">              <span v-text="patientObj.name" />            </el-form-item>            <el-form-item label="身份证号:">              <span v-text="patientObj.idCard" />            </el-form-item>            <el-form-item label="电话:">              <span v-text="patientObj.phone" />            </el-form-item>          </el-card>        </el-form>        <el-card style="margin-top:3px">          <div v-if="allPatientMsg.length>0">            <el-collapse accordion>              <el-collapse-item                v-for="(item,index) in allPatientMsg"                :key="index"                :name="index"              >                <template slot="title">                  科室:【{{ item.deptName }}】就诊时间:【{{ item.careDate }}】                </template>                <div>主诉:{{ item.caseTitle }}</div>                <div>诊断信息:{{ item.caseResult }}</div>                <div>医生建议:{{ item.doctorTips }}</div>                <div>备注:{{ item.remark }}</div>                <!--当前挂号单历史处方开始-->                <el-collapse v-show="item.careOrders.length>0" accordion>                  <el-collapse-item v-for="(it,i) in item.careOrders" :key="i">                    <template slot="title">                      【 {{ it.coType==='0'?'药品处方':'检查处方' }}】【{{ i+1 }}】 <span>【处方总额】:￥{{ it.allAmount }}</span>                    </template>                    <el-table                      v-loading="loading"                      border                      :data="it.careOrderItems"                    >                      <el-table-column label="序号" align="center" type="index" width="50" />                      <el-table-column :label="it.coType==='0'?'药品名称':'项目名称'" align="center" prop="itemName" />                      <el-table-column label="数量" align="center" prop="num" />                      <el-table-column label="单价(元)" align="center" prop="price" />                      <el-table-column label="金额(元)" align="center" prop="amount" />                      <el-table-column label="备注" prop="remark" align="center" />                      <el-table-column label="状态" prop="status" align="center" :formatter="statusFormatter" />                    </el-table>                  </el-collapse-item>                </el-collapse>              </el-collapse-item>            </el-collapse>          </div>          <div v-else style="text-align:center">            暂无就诊数据          </div>        </el-card>        <span slot="footer" class="dialog-footer">          <el-button @click="cancelHistory">取 消</el-button>        </span>      </el-dialog>      <!-- 病例信息弹出层结束 -->    </div>  </template>  <script>  // 引入api  import { listPatientForPage, getPatientById, getPatientFileById, getPatientAllMessageByPatientId } from '@/api/docter/patient'  export default {    // 定义页面数据    data() {      return {        // 是否启用遮罩层        loading: false,        // 选中数组        ids: [],        // 非单个禁用        single: true,        // 非多个禁用        multiple: true,        // 分页数据总条数        total: 0,        // 字典表格数据        patientTableList: [],        // 弹出层标题        title: '',        // 是否显示档案弹出层        fileOpen: false,        // 性别数据字典        sexOptions: [],        // 完善信息状态        isFinalOptions: [],        // 查询参数        queryParams: {          pageNum: 1,          pageSize: 10,          name: undefined,          idCard: undefined,          phone: undefined        },        // 患者信息        patientObj: {},        // 档案信息        patientFileObj: {},        // 是否打开病历弹出层        careHistoryOpen: false,        // 患者所有就诊信息        allPatientMsg: [],        // 订单状态        statusOptions: []      }    },    // 勾子    created() {      // 加载性别      this.getDataByType('sys\_user\_sex').then(res => {        this.sexOptions = res.data      })      // 加载性别      this.getDataByType('his\_patient\_msg\_final').then(res => {        this.isFinalOptions = res.data      })      //  加载处方详情支付状态      this.getDataByType('his\_order\_details\_status').then(res => {        this.statusOptions = res.data      })      // 查询表格数据      this.getPatientList()    },    // 方法    methods: {      // 查询表格数据      getPatientList() {        this.loading = true // 打开遮罩        listPatientForPage(this.addDateRange(this.queryParams, this.dateRange)).then(res => {          this.patientTableList = res.data          this.total = res.total          this.loading = false// 关闭遮罩        })      },      // 条件查询      handleQuery() {        this.getPatientList()      },      // 重置查询条件      resetQuery() {        this.resetForm('queryForm')        this.getPatientList()      },      // 数据表格的多选择框选择时触发      handleSelectionChange(selection) {        this.ids = selection.map(item => item.patientId)        this.single = selection.length !== 1        this.multiple = !selection.length      },      // 分页pageSize变化时触发      handleSizeChange(val) {        this.queryParams.pageSize = val        // 重新查询        this.getPatientList()      },      // 点击上一页  下一页，跳转到哪一页面时触发      handleCurrentChange(val) {        this.queryParams.pageNum = val        // 重新查询        this.getPatientList()      },      // 翻译性别      sexFormatter(row) {        return this.selectDictLabel(this.sexOptions, row.sex)      },      // 翻译信息状态      isFinalFormatter(row) {        return this.selectDictLabel(this.isFinalOptions, row.isFinal)      },      // 根据出生年月计算年龄      calculateAgeFormatter(row) {        return this.getAge(row.birthday.substring(0, 10))      },      // 查询患者档案      handleViewFile() {        const patientId = this.ids[0]        this.fileOpen = true        this.title = '查询患者档案信息'        this.patientFileObj = {}        this.patientObj = {}        // 查询患者        getPatientById(patientId).then(res => {          this.patientObj = res.data          const sex = this.patientObj.sex          const isFinal = this.patientObj.isFinal          this.patientObj.sex = (sex === '0' ? '男' : sex === '1' ? '女' : '未知')          this.patientObj.isFinal = isFinal === '0' ? '未完善' : '已完善'        })        // 查询档案        getPatientFileById(patientId).then(res => {          if (res.data !== null) {            this.patientFileObj = res.data          }        })        //      },      // 取消      cancelFile() {        this.fileOpen = false        this.title = ''      },      // 查询患者就诊信息      handleViewCareHistory(row) {        this.careHistoryOpen = true        const patientId = this.ids[0]        this.title = '查询患者病历信息'        this.patientFileObj = {}        this.patientObj = {}        // 查询患者        getPatientById(patientId).then(res => {          this.patientObj = res.data        })        // 查询病历        getPatientAllMessageByPatientId(patientId).then(res => {          this.loading = false          this.allPatientMsg = res.data        }).catch(() => {          this.msgError('查询失败')        })      },      // 取消按钮      cancelHistory() {        this.careHistoryOpen = false      },      // 翻译处方详情支付状态类型      statusFormatter(row, column) {        return this.selectDictLabel(this.statusOptions, row.status)      }    }  }  </script>  <style scoped>    .demo-table-expand {      font-size: 0;    }    .demo-table-expand label {      width: 90px;      color: #99a9bf;    }    .demo-table-expand .el-form-item {      margin-right: 0;      margin-bottom: 0;      width: 50%;    }  </style> |