一、首页数据显示

1、医院列表显示（接口）

1.1 service接口与实现

在管理平台 医院分页列表时已经提供，目前我们可以直接使用

1.2 添加controller方法

创建HospitalApiController

```
@Api(tags = "医院显示接口")
@RestController
@RequestMapping("/api/hosp/hospital")
public class HospitalApiController {

    @Autowired
    private HospitalService hospitalService;

    @ApiOperation(value = "获取分页列表")
    @GetMapping("{page}/{limit}")
    public R index(
            @PathVariable Integer page,
            @PathVariable Integer limit,
            HospitalQueryVo hospitalQueryVo) {

        Page<Hospital> pageModel = hospitalService.selectHospPage(page, limit, hospit
        return R.ok().data("pages",pageModel);
    }
}
```

2、医院搜索（接口）

2.1 service接口与实现

在HospitalService添加接口和实现

```
/**
     * 根据医院名称获取医院列表
     */
List<Hospital> findByHosname(String hosname);

@Override
public List<Hospital> findByHosname(String hosname) {
    return hospitalRepository.findHospitalByHosnameLike(hosname);
}
```

在HospitalRepository类添加方法

```
List<Hospital> findHospitalByHosnameLike(String hosname);
```

2.2 .HospitalApiController 添加方法

```
@ApiOperation(value = "根据医院名称获取医院列表")
@GetMapping("findByHosname/{hosname}")
public R findByHosname(
    @ApiParam(name = "hosname", value = "医院名称", required = true)
    @PathVariable String hosname) {

    List<Hospital> list = hospitalService.findByHosname(hosname);
    return R.ok().data("list",list);
}
```

3、首页数据前端

3.1 封装api

（1）创建api文件夹，创建/api/yygh/hospital.js

```
import request from '@/utils/request'

const api_name = `/api/hosp/hospital`

export default {
    getPageList(page, limit, searchObj) {
        return request({
            url: `${api_name}/${page}/${limit}`,
            method: 'get',
            params: searchObj
        })
    },
    getByHosname(hosname) {
        return request({
            url: `${api_name}/findByHosname/${hosname}`,
            method: 'get'
        })
    }
}
```

（2）创建/api/yygh/dict.js

```
import request from '@/utils/request'

const api_name = '/admin/cmn/dict'

export default {
    findByDictCode(dictCode) {
        return request({
            url: `${api_name}/findByDictCode/${dictCode}`,
            method: 'get'
        })
    },
    findByParentId(parentId) {
        return request({
            url: `${api_name}/findChildData/${parentId}`,
            method: 'get'
        })
    }
}
```

3.2 添加组件

（1）修改pages/index.vue组件

```
<template>
  <div class="home page-component">
    <el-carousel indicator-position="outside">
      <el-carousel-item v-for="item in 2" :key="item">
        <img src="~assets/images/web-banner1.png" alt="">
      </el-carousel-item>
    </el-carousel>
    <!-- 搜索 -->
    <div class="search-container">
    <div class="search-wrapper">
    <div class="hospital-search">
      <el-autocomplete
      class="search-input"
      prefix-icon="el-icon-search"
      v-model="hosname"
      :fetch-suggestions="querySearchAsync"
      placeholder="点击输入医院名称"
      @select="handleSelect"
      >
        <span slot="suffix" class="search-btn v-link highlight clickable selected">搜
      </el-autocomplete>
    </div>
    </div>
    </div>
    <!-- bottom -->
    <div class="bottom">
    <div class="left">
    <div class="home-filter-wrapper">
    <div class="title"> 医院</div>
    <div>
      <div class="filter-wrapper">
        <span
        class="label">等级：</span>
        <div class="condition-wrapper">
          <span class="item v-link clickable"
```

```
            :class="hostypeActiveIndex == index ? 'selected' : ''"
             v-for="(item,index) in hostypeList" :key="item.id"
             @click="hostypeSelect(item.value, index)">{{ item.name }}</span>
       </div>
      </div>
    <div class="filter-wrapper">
      <span
      class="label">地区：</span>
      <div class="condition-wrapper">
        <span class="item v-link clickable"
          :class="provinceActiveIndex == index ? 'selected' : ''"
          v-for="(item,index) in districtList" :key="item.id"
          @click="districtSelect(item.value, index)">{{ item.name }}</span>
      </div>
      </div>
    </div>
    </div>
    <div class="v-scroll-list hospital-list">
      <div class="v-card clickable list-item" v-for="item in list" :key="item.id">
        <div class="">
          <div class="hospital-list-item hos-item" index="0" @click="show(item.hosco
            <div class="wrapper">
            <div class="hospital-title"> {{ item.hosname }}</div>
            <div class="bottom-container">
            <div class="icon-wrapper">
              <span class="iconfont">□</span>{{ item.param.hostypeString }}
            </div>
          <div class="icon-wrapper">
          <span class="iconfont">□</span>每天{{ item.bookingRule.releaseTime }}放号
          </div>
          </div>
          </div>
          <img :src="'data:image/jpeg;base64,'+item.logoData"
           :alt="item.hosname"
           class="hospital-img">
      </div>
    </div>
    </div>
    </div>
    </div>
    <div class="right">
      <div class="common-dept">
      <div class="header-wrapper">
      <div class="title"> 常见科室</div>
      <div class="all-wrapper"><span>全部</span>
      <span class="iconfont icon">□</span>
      </div>
      </div>
      <div class="content-wrapper">
      <span class="item v-link clickable dark">神经内科 </span>
      <span class="item v-link clickable dark">消化内科 </span>
      <span class="item v-link clickable dark">呼吸内科 </span>
      <span class="item v-link clickable dark">内科 </span>
      <span class="item v-link clickable dark">神经外科 </span>
      <span class="item v-link clickable dark">妇科 </span>
      <span class="item v-link clickable dark"> 产科 </span>
      <span class="item v-link clickable dark">儿科 </span>
      </div>
    </div>
    <div class="space">
      <div class="header-wrapper">
      <div class="title-wrapper">
      <div class="icon-wrapper"><span
      class="iconfont title-icon">□</span>
      </div>
      <span class="title">平台公告</span>
      </div>
      <div class="all-wrapper">
      <span>全部</span>
      <span class="iconfont icon">□</span>
      </div>
      </div>
      <div class="content-wrapper">
      <div class="notice-wrapper">
      <div class="point"></div>
      <span class="notice v-link clickable dark">关于延长北京大学国际医院放假的通知 </sp
      </div>
      <div class="notice-wrapper">
      <div class="point"></div>
      <span class="notice v-link clickable dark">北京中医药大学东方医院部分科室医生门诊
      </div>
      <div class="notice-wrapper">
      <div class="point"></div>
      <span class="notice v-link clickable dark"> 武警总医院号源暂停更新通知 </span>
      </div>
      </div>
    </div>
    <div class="suspend-notice-list space">
    <div class="header-wrapper">
    <div class="title-wrapper">
      <div class="icon-wrapper">
      <span class="iconfont title-icon">□</span>
      </div>
      <span class="title">停诊公告</span>
      </div>
      <div class="all-wrapper">
      <span>全部</span>
      <span class="iconfont icon">□</span>
      </div>
      </div>
      <div class="content-wrapper">
      <div class="notice-wrapper">
      <div class="point"></div>
      <span class="notice v-link clickable dark"> 中国人民解放军总医院第六医学中心(原海
      </div>
      <div class="notice-wrapper">
      <div class="point"></div>
    <span class="notice v-link clickable dark"> 首都医科大学附属北京潞河医院老年医学科门
    </div>
      <div class="notice-wrapper">
        <div class="point"></div>
        <span class="notice v-link clickable dark">中日友好医院中西医结合心内科门诊停诊
      </div>
    </div>
    </div>
    </div>
    </div>
  </div>
</template>
<script>

import hospApi from '@/api/hosp/hospital.js'
import dictApi from '@/api/hosp/dict.js'

export default {
  //服务端渲染异步，显示医院列表
  asyncData({ params, error }) {
    //调用
    return hospApi.getPageList(1,10,null)
```

```
      .then(response => {
        return {
          list: response.data.pages.content,
          pages: response.data.pages.totalPages
        }
      })
  },
  data() {
    return {
      searchObj: {},
      page: 1,
      limit: 10,

      hosname: '', //医院名称
      hostypeList: [], //医院等级集合
      districtList: [], //地区集合

      hostypeActiveIndex: 0,
      provinceActiveIndex: 0
    }
  },
  created() {
    this.init()
  },
  methods:{
    //查询医院等级列表 和 所有地区列表
    init() {
      //查询医院等级列表
      dictApi.findByDictCode('Hostype')
        .then(response => {
          //hostypeList清空
          this.hostypeList = []
          //向hostypeList添加全部值
          this.hostypeList.push({"name":"全部","value":""})
          //把接口返回数据，添加到hostypeList
          for(var i=0;i<response.data.list.length;i++) {
              this.hostypeList.push(response.data.list[i])
          }
      })

      //查询地区数据
      dictApi.findByDictCode('Beijin')
        .then(response => {
          this.districtList = []
          this.districtList.push({"name":"全部","value":""})
          for(let i in response.data.list) {
            this.districtList.push(response.data.list[i])
          }
        })
    },

    //查询医院列表
    getList() {
      hospApi.getPageList(this.page,this.limit,this.searchObj)
        .then(response => {
          for(let i in response.data.pages.content) {
            this.list.push(response.data.pages.content[i])
          }
          this.page = response.data.pages.totalPages
        })
    },

    //根据医院等级查询
    hostypeSelect(hostype,index) {
      //准备数据
      this.list = []
      this.page = 1
      this.hostypeActiveIndex = index
      this.searchObj.hostype = hostype
      //调用查询医院列表方法
      this.getList()
    },

    //根据地区查询医院
    districtSelect(districtCode, index) {
      this.list = []
      this.page = 1
      this.provinceActiveIndex = index
      this.searchObj.districtCode = districtCode
      this.getList();
    },

    //在输入框输入值，弹出下拉框，显示相关内容
    querySearchAsync(queryString, cb) {
      this.searchObj = []
      if(queryString == '') return
```

```
   hospApi.getByHosname(queryString).then(response => {
    for (let i = 0, len = response.data.list.length; i <len; i++) {
     response.data.list[i].value = response.data.list[i].hosname
    }
    cb(response.data.list)
   })
  },

  //在下拉框选择某一个内容，执行下面方法，跳转到详情页面中
  handleSelect(item) {
   window.location.href = '/hospital/' + item.hoscode
  },

  //点击某个医院名称，跳转到详情页面中
  show(hoscode) {
   window.location.href = '/hospital/' + hoscode
  }
 }
}
</script>
```

3.3 修改utils里面request.js路径

修改utils/request.js文件

```
// 请求地址端口号修改为 gateway网关地址
const service = axios.create({
  baseURL: 'http://localhost:8222/',
  timeout: 15000 // 请求超时时间
})

...................

// http response 拦截器
// 返回状态码改为 20000
service.interceptors.response.use(
  response => {
    if (response.data.code !== 20000) {

...................
```