

优医问诊day04

首页、极速问诊



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

01

首页模块-搭建页面

首页模块-搭建页面



组件基础结构

知识列表Tab

```
<!-- 头部 -->
<div class="home-header"> ...
</div>
<!-- 导航 -->
<div class="home-navs"> ...
</div>
<!-- 轮播图 -->
<div class="home-banner"> ...
</div>
```

```
<script setup lang="ts">
import { ref } from 'vue'

const active = ref(1)
</script>

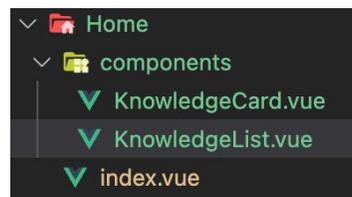
<template>
<div class="home-page">
  <!-- TAB -->
  <van-tabs shrink sticky v-model:active="active">
    <van-tab title="关注"></van-tab>
    <van-tab title="推荐">
      <p v-for="i in 100" :key="i">内容</p>
    </van-tab>
    <van-tab title="减脂"></van-tab>
    <van-tab title="饮食"></van-tab>
  </van-tabs>
</div>
</template>
```

```
// 全局覆盖van-tab样式
.van-tabs {
  .van-tabs__nav {
    padding: 0 0 15px 0;
  }
  .van-tabs__line {
    width: 20px;
    background-color: var(--cp-primary);
  }
  .van-tab {
    padding: 0 15px;
  }
}
```

02

首页模块-列表加载更多效果

首页模块-列表加载更多效果



```
<van-list
  v-model:loading="loading"
  @load="onLoad"
  :finished="finished"
  finished-text="没有更多了"
>
```

```
const loading = ref(false)
const finished = ref(false)
const list = ref<number[]>([])
const onLoad = () => {
  // 模拟发请求
  const data = [1, 2, 3, 4, 5]
  setTimeout(() => {
    list.value.push(...data)
    // 模拟加载完毕
    if (list.value.length > 20) {
      finished.value = true
    }
    loading.value = false
  }, 2000)
}
```

```
<knowledge-card v-for="(item, i) in list" :key="i"></knowledge-card>
```

03

首页模块-知识列表数据类型

首页模块-知识列表数据类型

返回响应

成功 (200)

HTTP 状态码: 200 内容格式: JSON

数据结构

- success** boolean
请求成功标志
- code** number
请求成功10000标志
- message** string
请求成功
- data** object (3)
 - 返回数据
 - pageTotal** number
总页数
 - total** number
总条数
 - rows** array [object {14}]
数据

响应数据类型

请求参数类型

props类型

```
// 文章信息类型
export type Knowledge = {
  id: string
  title: string
  coverUrl: string[]
  topics: string[]
  collectionNumber: number
  commentNumber: number
  creatorName: string
  creatorAvatar: string
  creatorHospitalName: string
  likeFlag: 0 | 1
  content: string
  creatorDep: string
  creatorTitles: string
  creatorId: string
}

// 文章列表
export type KnowledgeList = Knowledge[]

// 文章列表带分页
export type KnowledgePage = {
  pageTotal: number
  total: number
  rows: KnowledgeList
}
```

```
// props类型 recommend推荐, fatReduction减脂, food健康饮食, like关注医生页面文章
export type KnowledgeType = 'like' | 'recommend' | 'fatReduction' | 'food'

// 文章列表查询参数
export type KnowledgeParams = {
  type: KnowledgeType
  current: number
  pageSize: number
}
```

```
defineProps<{
  type: KnowledgeType
}>()
```

```
<van-tab title="关注">
  <knowledge-list type="like"></knowledge-list>
</van-tab>
<van-tab title="推荐">
  <knowledge-list type="recommend"></knowledge-list>
</van-tab>
<van-tab title="减脂">
  <knowledge-list type="fatReduction"></knowledge-list>
</van-tab>
<van-tab title="饮食">
  <knowledge-list type="food"></knowledge-list>
</van-tab>
```

请求参数

参数名	位置	类型	必填	说明
type	query	string	是	示例值: like recommend推荐, fatReduction减脂, food健康饮食like关注医生页面文章
current	query	string	是	示例值: 1
pageSize	query	string	是	示例值: 10

04

首页模块-知识列表加载功能

首页模块-知识列表加载功能



```
export const getKnowledgePage = (params: KnowledgeParams) =>
  request<KnowledgePage>('/patient/home/knowledge', 'GET', params)
```

```
const props = defineProps<{
  type: KnowledgeType
}>()

const loading = ref(false)
const finished = ref(false)

const list = ref<KnowledgeList>([])
const params = ref<KnowledgeParams>({
  type: props.type,
  current: 1,
  pageSize: 10
})
```

```
<knowledge-card
  v-for="item in list"
  :key="item.id"
  :item="item"
></knowledge-card>
```

```
<p class="intro van-multi-ellipsis--l2">
  {{ item.content.replace(/<[>]+>/g, '') }}
</p>
<div class="imgs" :class="{ large: item.coverUrl.length === 1 }">
  <van-image
    fit="cover"
    v-for="(url, i) in item.coverUrl"
    :key="i"
    :src="url"
  />
</div>
```

```
const onLoad = async () => {
  // 加载更多
  const res = await getKnowledgePage(params.value)
  list.value.push(...res.data.rows)
  if (params.value.current >= res.data.pageTotal) {
    finished.value = true
  } else {
    params.value.current++
  }
  loading.value = false
}
```

05

首页模块-搭建推荐关注组件

首页模块-搭建推荐关注组件



```
<div class="follow-doctor">
  <div className="head">
    <p>推荐关注</p>
    <a href="javascript:;"> 查看更多<i class="van-icon van-icon-arrow" /></a>
  </div>
  <div class="body">
    <!-- swipe 组件 -->
    <van-swipe>
      <van-swipe-item v-for="item in 5" :key="item">
        {{ item }}
      </van-swipe-item>
    </van-swipe>
  </div>
</div>
```

```
<!-- swipe 组件 -->
<van-swipe :showIndicators="false" :loop="false" :width="150">
```

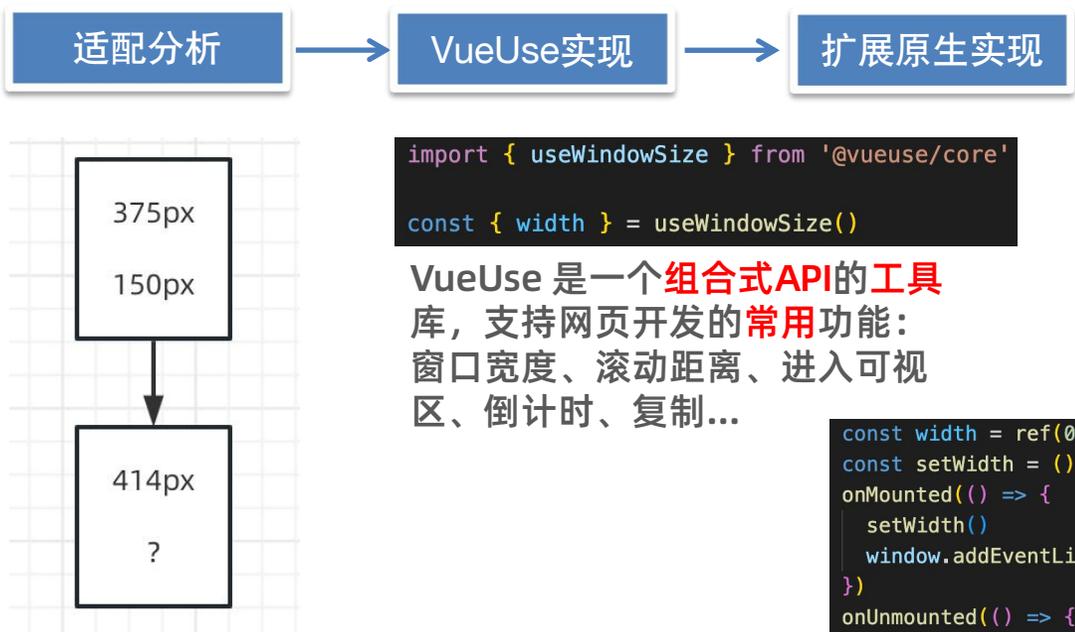
```
<template>
  <div class="doctor-card">
    <van-image
      round
      src="https://yanxuan-item.nosdn.127.net/3cb61b3fd4761555e56c4a5f19d1b4b1.png"
    />
    <p class="name">周医生</p>
    <p class="van-ellipsis">积水潭医院 神经内科</p>
    <p>副主任医师</p>
    <van-button round size="small" type="primary">+ 关注</van-button>
  </div>
</template>
```

```
<van-swipe-item v-for="item in 5" :key="item">
  <doctor-card />
</van-swipe-item>
```

06

首页模块-实现适配-利用VueUse

首页模块-实现适配-利用VueUse



```
import { useWindowSize } from '@vueuse/core'
const { width } = useWindowSize()
```

VueUse 是一个**组合式API的工具**库，支持网页开发的**常用功能**：窗口宽度、滚动距离、进入可视区、倒计时、复制...

```
const width = ref(0)
const setWidth = () => (width.value = window.innerWidth)
onMounted(() => {
  setWidth()
  window.addEventListener('resize', setWidth)
})
onUnmounted(() => {
  window.removeEventListener('resize', setWidth)
})
```

07

首页模块-推荐医生展示

首页模块-推荐医生展示



```
// 通用的分页查询参数
export type PageParams = {
  // 当前页码
  current: number
  // 每页条数
  pageSize: number
}

// 文章列表查询参数
export type KnowledgeParams = PageParams & {
  // 文章类型
  type: KnowledgeType
}

// 医生卡片对象
export type Doctor = { ... }

// 医生列表
export type DoctorList = Doctor[]

// 医生分页
export type DoctorPage = {
  pageTotal: number
  total: number
  rows: DoctorList
}
```

```
export const getDoctorPage = (params: PageParams) =>
  request<DoctorPage>('/home/page/doc', 'GET', params)
```

```
const list = ref<DoctorList>()
const loadData = async () => {
  const res = await getDoctorPage({ current: 1, pageSize: 5 })
  list.value = res.data.rows
}
onMounted(() => loadData())
```

```
<van-swipe-item v-for="item in list" :key="item.id">
  <doctor-card :item="item"></doctor-card>
</van-swipe-item>
```

```
<script lang="ts" setup>
import type { Doctor } from '@/types/consult'

defineProps<{ item: Doctor }>()
</script>
<template>
  <div class="doctor-card">
    <van-image round :src="item.avatar" />
    <p class="name">{{ item.name }}</p>
    <p class="van-ellipsis">{{ item.hospitalName }} {{ item.depName }}</p>
    <p>{{ item.positionalTitles }}</p>
    <van-button round size="small" type="primary">
      {{ item.likeFlag === 1 ? '已关注' : '+ 关注' }}
    </van-button>
  </div>
</template>
```

08

首页模块-实现关注医生

首页模块-实现关注医生



关注TS类型

关注API

关注&取消

```
// 关注的类型, 医生|文章|百科话题|疾病  
export type FollowType = 'doc' | 'knowledge' | 'topic' | 'disease'
```

```
export const followOrUnfollow = (id: string, type: FollowType = 'doc') =>  
  request('/like', 'POST', { id, type })
```

```
// 关注医生逻辑  
const loading = ref(false)  
const follow = async (doc: Doctor) => {  
  loading.value = true  
  try {  
    await followOrUnfollow(doc.id)  
    doc.likeFlag = doc.likeFlag === 1 ? 0 : 1  
  } finally {  
    loading.value = false  
  }  
}
```

```
<van-button  
  :loading="loading"  
  @click="follow(item)"  
  round  
  size="small"  
  type="primary"  
>  
  {{ item.likeFlag === 1 ? '已关注' : '+ 关注' }}  
</van-button>
```

09

首页模块-composables组合函数

首页模块-逻辑复用



封装关注逻辑

实现关注医生

实现关注文章

```
// 封装逻辑 composable , 规范 useXxx, 表示使用某功能
export const useFollow = (type: FollowType = 'doc') => {
  const loading = ref(false)
  const follow = async (item: { id: string; likeFlag: 0 | 1 }) => {
    loading.value = true
    try {
      await followOrUnfollow(item.id, type)
      item.likeFlag = item.likeFlag === 1 ? 0 : 1
    } finally {
      loading.value = false
    }
  }
  return { loading, follow }
}
```

```
// 关注知识文章
const { loading, follow } = useFollow('knowledge')
```

```
<van-button
  class="btn"
  size="small"
  round
  :loading="loading"
  @click="follow(item)"
>
  {{ item.likeFlag === 1 ? '已关注' : '+ 关注' }}
</van-button>
```

```
// 关注医生
const { loading, follow } = useFollow()
```

在 Vue 应用的概念中，“组合式函数”(Composables) 是一个利用 Vue 的组合式 API 来封装和复用有状态逻辑的函数。

10

极速问诊-流程分析

极速问诊-流程分析



11

极速问诊-TypeScript 枚举

极速问诊-TypeScript 枚举



问医生

按科室查问医生



极速问诊

20s医生极速回复



开药门诊

线上买药更方便



```
type number
```

就诊类型1找医生 2极速问诊 3开药问诊默认是1

```
type Obj = {  
  type: 1 | 2 | 3  
}
```

语义差
建议使用枚举

TypeScript 枚举允许开发人员定义一组命名常量，使用枚举可以更容易的记录意图。

```
// 方向: 0 1 2 3  
enum Direction {  
  Up = 0,  
  Down = 1,  
  Left = 2,  
  Right = 3  
}  
  
// 改变方向  
const changeDirection = (direction: Direction) => {  
  console.log(direction)  
}  
  
// 使用枚举  
changeDirection(Direction.Right)
```

建议枚举定义在ts文件中，因为它也参与运行



思考

1. 在 topic knowledge doc disease 这组值内可选?
字面量联合类型
2. 在 1 2 3 这组值内可选, 1问医生2极速3开药?
枚举类型

12

极速问诊-问诊信息类型定义

极速问诊-问诊信息类型定义



```
// 问诊类型
export enum ConsultType {
  /** 找医生 */
  Doctor = 1,
  /** 快速问诊 */
  Fast = 2,
  /** 开药问诊 */
  Medication = 3
}
```

<code>type number</code>	就诊类型 1找医生 2极速问诊 3开药问诊默认是1
<code>couponId string</code>	使用优惠券, 必传可使用优惠券id
<code>illnessType integer</code>	极速问诊类型: 0普通 1三甲
<code>patientId string</code>	患者id,用于关联患者信息
<code>depId string</code>	选择的科室id-极速问诊必填(选择的科室)
<code>illnessDesc string</code>	病情--症状描述
<code>illnessTime number</code>	找医生/极速问诊-患病时间 1一周内 2一月内 3半年内 4半年以上
<code>consultFlag number</code>	找医生/极速问诊-是否就诊过 0未就诊 1就诊过
<code>pictures array [object {2}]</code>	补充病例信息-图片集合

```
// 问诊时间, 以1自增可以省略
export enum IllnessTime {
  /** 一周内 */
  Week = 1,
  /** 一月内 */
  Month,
  /** 半年内 */
  HalfYear,
  /** 半年以上 */
  More
}
```

```
// 图片列表
export type Image = {
  /** 图片ID */
  id: string
  /** 图片地址 */
  url: string
}
// 问诊记录
export type Consult = {
  /** 问诊记录ID */
  id: string
  /** 问诊类型 */
  type: ConsultType
  /** 快速问诊类型, 0 普通 1 三甲 */
  illnessType: 0 | 1
  /** 科室ID */
  depId: string
  /** 疾病描述 */
  illnessDesc: string
  /** 疾病持续时间 */
  illnessTime: IllnessTime
  /** 是否就诊过, 0 未就诊过 1 就诊过 */
  consultFlag: 0 | 1
  /** 图片数组 */
  pictures: Image[]
  /** 患者ID */
  patientId: string
  /** 优惠券ID */
  couponId: string
}
```

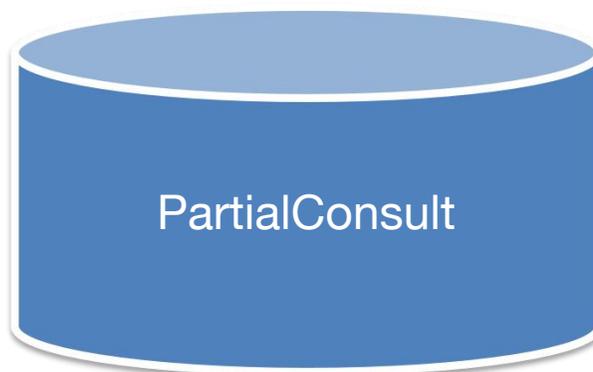
```
// 问诊记录-全部可选
export type PartialConsult = Partial<Consult>
```

TypeScript 内置类型:
Required<T> 对象属性全部**必选**
Partial<T> 对象属性全部**可选**

13

极速问诊-选择问诊类型

极速问诊-选择问诊类型



```
/** 问诊类型 */  
type: ConsultType  
/** 快速问诊类型, 0 普通 1 三甲 */  
illnessType: 0 | 1  
/** 科室ID */  
depId: string  
/** 病情描述 */  
illnessDesc: string  
/** 疾病持续时间 */  
illnessTime: IllnessTime  
/** 是否就诊过, 0 未就诊过 1 就诊过 */  
consultFlag: 0 | 1  
/** 图片数组 */  
pictures: Image[]  
/** 患者ID */  
patientId: string  
/** 优惠券ID */  
couponId: string
```

极速问诊-选择问诊类型



定义问诊仓库

提供记录函数

记录问诊类型

```
export const useConsultStore = defineStore(  
  'cp-consult',  
  () => {  
    const consult = ref<PartialConsult>({})  
    // 设置问诊类型  
    const setType = (type: ConsultType) => (consult.value.type = type)  
  
    return { consult, setType }  
  },  
  {  
    persist: true  
  }  
)
```

```
<router-link  
  to="/consult/fast"  
  @click="store.setType(ConsultType.Fast)"  
  class="nav"  
>
```

14

极速问诊-选择问诊级别

极速问诊-选择问诊级别



```
{  
  path: '/consult/fast',  
  component: () => import('@/views/Consult/ConsultFast.vue'),  
  meta: { title: '极速问诊' }  
},
```

```
// 设置极速问诊类型  
const setIllnessType = (type: 0 | 1) => (consult.value.illnessType = type)
```

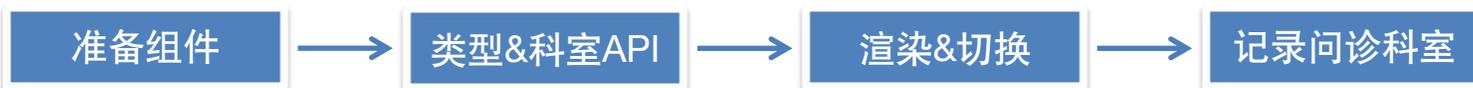
```
<router-link  
  to="/consult/dep"  
  class="item"  
  @click="store.setIllnessType(1)"  
>  
  <cp-icon class="pic" name="consult-doctor"></cp-icon>  
  <div class="info">  
    <p>三甲图文问诊</p>  
    <p>三甲主治及以上级别医生</p>  
  </div>  
  <van-icon name="arrow"></van-icon>  
</router-link>  
<router-link  
  to="/consult/dep"  
  class="item"  
  @click="store.setIllnessType(0)"  
>  
  <cp-icon class="pic" name="consult-message"></cp-icon>  
  <div class="info">  
    <p>普通图文问诊</p>  
    <p>二甲主治及以上级别医生</p>  
  </div>  
  <van-icon name="arrow"></van-icon>  
</router-link>
```

15

极速问诊-选择问诊科室

极速问诊-选择问诊科室

选择科室	
内科	心血管内科
外科	普通内科
妇产科学	神经内科
儿科学	消化内科
骨外科	内分泌科
眼科学	免疫科
口腔科学	高压氧科
耳鼻咽喉头颈科	血液科
肿瘤科	肾病内科
皮肤性病科	呼吸科
男科	感染内科
皮肤美容	过敏反应科
修作科	



```
export const getAllDep = () => request<TopDep[]>('/dep/all')
```

```
{
  path: '/consult/dep',
  component: () => import('@/views/Consult/ConsultDep.vue'),
  meta: { title: '选择科室' }
},
```

```
<div class="consult-dep-page">
  <cp-nav-bar title="选择科室" />
  <div class="wrapper">
    <van-sidebar v-model="active">
      <van-sidebar-item title="内科" />
      <van-sidebar-item title="外科" />
      <van-sidebar-item title="皮肤科" />
      <van-sidebar-item title="骨科" />
    </van-sidebar>
    <div class="sub-dep">
      <router-link to="/consult/illness">科室一</router-link>
      <router-link to="/consult/illness">科室二</router-link>
      <router-link to="/consult/illness">科室三</router-link>
    </div>
  </div>
</div>
```

```
// 一级科室
const allDep = ref<TopDep[]>([])
onMounted(async () => {
  const res = await getAllDep()
  allDep.value = res.data
})

// 二级科室, 注意: 组件初始化没有数据 child 可能拿不到
const subDep = computed(() => allDep.value[active.value]?.child)
```

```
<div class="wrapper">
  <van-sidebar v-model="active">
    <van-sidebar-item
      :title="top.name"
      v-for="top in allDep"
      :key="top.id"
    />
  </van-sidebar>
  <div class="sub-dep">
    <router-link
      to="/consult/illness"
      v-for="sub in subDep"
      :key="sub.id"
      @click="store.setDep(sub.id)"
    >
      {{ sub.name }}
    </router-link>
  </div>
</div>
```

```
// 设置科室
const setDep = (id: string) => (consult.value.depId = id)
```

```
<div class="sub-dep">
  <router-link
    to="/consult/illness"
    v-for="sub in subDep"
    :key="sub.id"
    @click="store.setDep(sub.id)"
  >
    {{ sub.name }}
  </router-link>
</div>
```



传智教育旗下高端IT教育品牌