

优医问诊day08

药品订单、三方登录



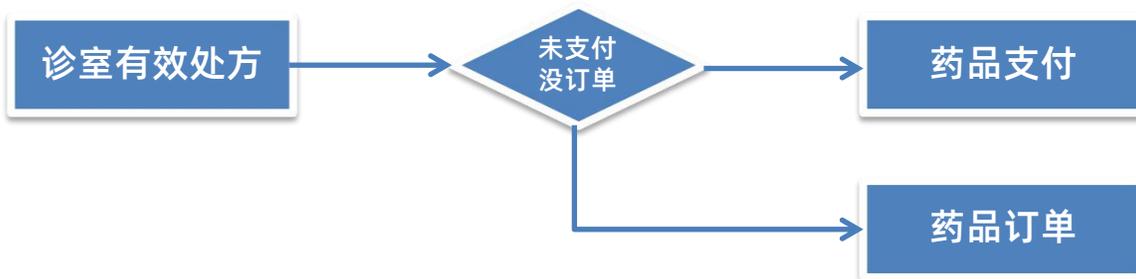
黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

01

药品订单-进入药品支付

药品订单-进入药品支付



```
// 跳转买药
const router = useRouter()
const buy = (pre?: Prescription) => {
  if (!pre) return
  // 处方失效
  if (pre.status === PrescriptionStatus.Invalid) return showToast('处方已失效')
  // 没有生成药品订单
  if (pre.status === PrescriptionStatus.NotPayment && !pre.orderId)
    return router.push(`/order/pay?id=${pre.id}`)
  // 已生成过药品订单
  router.push(`/order/${pre.orderId}`)
}
```

```
{
  path: '/order/pay',
  component: () => import('@/views/Order/OrderPay.vue'),
  meta: { title: '药品支付' }
},
```



02

药品订单-支付信息

药品订单-支付信息



```
export type OrderPre = {  
  /** 处方ID */  
  id: string  
  /** 优惠券ID */  
  couponId: string  
  /** 积分抵扣 */  
  pointDeduction: number  
  /** 优惠券抵扣 */  
  couponDeduction: number  
  /** 应付款 */  
  payment: number  
  /** 邮费 */  
  expressFee: number  
  /** 实付款 */  
  actualPayment: number  
  /** 药品订单 */  
  medicines: Medical[]  
}
```

```
export type AddressItem = {  
  /** 地址ID */  
  id: string  
  /** 联系方式 */  
  mobile: string  
  /** 收件人 */  
  receiver: string  
  /** 省 */  
  province: string  
  /** 市 */  
  city: string  
  /** 区 */  
  county: string  
  /** 详细地址 */  
  addressDetail: string  
  /** 是否默认地址, 0 不是 1 是 */  
  isDefault: 0 | 1  
}
```

```
// 药品预支付信息  
export const getMedicalOrderPre = (params: { prescriptionId: string }) =>  
  request<OrderPre>('/patient/medicine/order/pre', 'GET', params)  
  
// 收货地址列表  
export const getAddressList = () =>  
  request<AddressItem[]>('/patient/order/address')
```

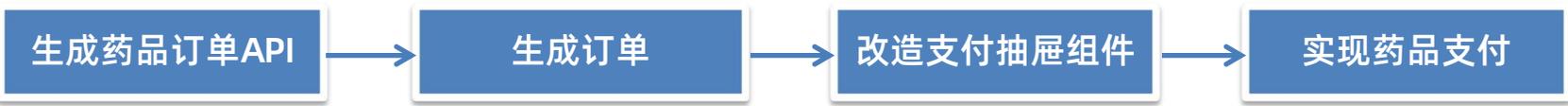
```
const route = useRoute()  
// 预支付信息  
const orderPre = ref<OrderPre>()  
const loadOrderPre = async () => {  
  const res = await getMedicalOrderPre({  
    prescriptionId: route.query.id as string  
  })  
  orderPre.value = res.data  
}  
  
// 收货地址  
const address = ref<AddressItem>()  
const loadAddress = async () => {  
  const addRes = await getAddressList()  
  if (addRes.data.length) {  
    const defAddress = addRes.data.find((item) => item.isDefault === 0)  
    if (defAddress) address.value = defAddress  
    else address.value = addRes.data[0]  
  }  
}  
  
onMounted(async () => {  
  loadOrderPre()  
  loadAddress()  
})
```

```
<div class="order-pay-page" v-if="orderPre && address">  
  <cp-nav-bar title="药品支付" />  
  <div class="order-address">--  
  </div>  
  <div class="order-medical">--  
  </div>  
  <div class="order-detail">--  
  </div>  
  <div class="order-tip">--  
  </div>  
  <van-submit-bar --  
  ></van-submit-bar>  
</div>  
  
<div class="order-pay-page" v-else>  
  <cp-nav-bar title="药品支付" />  
  <van-skeleton title avatar row="2" style="margin-top: 15px" />  
  <van-skeleton title row="4" style="margin-top: 50px" />  
  <van-skeleton title row="4" style="margin-top: 50px" />  
</div>
```

03

药品订单-完成支付

药品订单-完成支付



```
// 创建药品订单
export const createMedicalOrder = (data: {
  id: string
  addressId: string
  couponId?: string
}) => request<{ id: string }>('/patient/medicine/order', 'POST', data)
```

```
const agree = ref(false)
const loading = ref(false)
const orderId = ref('')
const show = ref(false)
const onSubmit = async () => {
  if (!agree.value) return showToast('请同意支付协议')
  if (!address.value?.id) return showToast('请选择收货地址')
  if (!orderPre.value?.id) return showToast('未找到处方')
  if (!orderId.value) {
    try {
      loading.value = true
      const res = await createMedicalOrder({
        id: orderPre.value.id,
        addressId: address.value.id,
        couponId: orderPre.value.couponId
      })
      loading.value = false
      orderId.value = res.data.id
      // 生成订单后打开支付抽屉
      show.value = true
    } catch (error) {
      loading.value = false
    }
  } else {
    // 已生成订单直接打开
    show.value = true
  }
}
```

```
const props = defineProps<{
  show: boolean
  orderId: string
  actualPayment: number
  onClose?: () => void
  payCallback: string
}>()
```

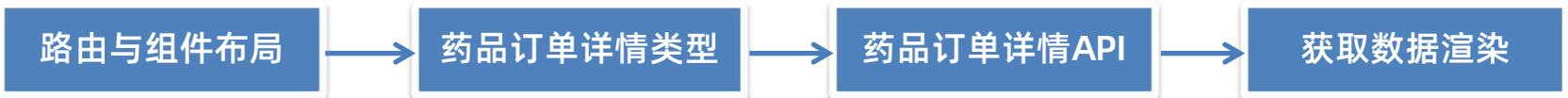
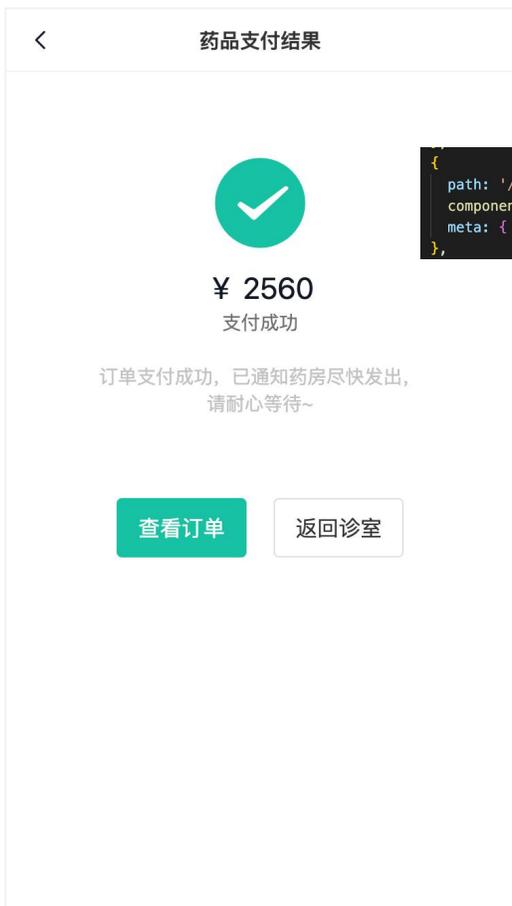
```
const res = await getConsultOrderPayUrl({
  paymentMethod: paymentMethod.value,
  orderId: props.orderId,
  payCallback: 'http://localhost:5173' + props.payCallback
})
```

```
<cp-pay-sheet
  v-model:show="show"
  :actual-payment="orderPre.actualPayment"
  :order-id="orderId"
  pay-callback="/order/pay/result"
></cp-pay-sheet>
```

04

药品订单-支付结果

药品订单-支付信息



```
{  
  path: '/order/pay/result',  
  component: () => import('@/views/Order/OrderPayResult.vue'),  
  meta: { title: '药品支付结果' }  
},
```

```
type Address = Omit<AddressItem, 'isDefault'>  
  
export type OrderDetail = {  
  /** 药品订单ID */  
  id: string  
  /** 药品订单编号 */  
  orderNo: string  
  /** 订单类型 */  
  type: number  
  /** 创建时间 */  
  createTime: string  
  /** 处方ID */  
  prescriptionId: string  
  /** 订单状态 */  
  status: OrderType  
  /** 订单状态说明 */  
  statusValue: string  
  /** 药品清单 */  
  medicines: Medical[]  
  /** 支付倒计时时间 */  
  countdown: number  
  /** 收货地址 */  
  addressInfo: Address  
  /** 物流信息 */  
  expressInfo: {  
    /** 物流最新位置 */  
    content: string  
    /** 物流最新时间 */  
    time: string  
  }  
  /** 支付时间 */  
  payTime: string  
  /** 支付方式 */  
  paymentMethod?: 0 | 1  
  /** 支付金额 */  
  payment: number  
  /** 积分抵扣 */  
  pointDeduction: number  
  /** 优惠券抵扣 */  
  couponDeduction: number  
  /** 邮费 */  
  expressFee: number  
  /** 实付金额 */  
  actualPayment: number  
  /** 问诊室ID */  
  roomId: string  
}
```

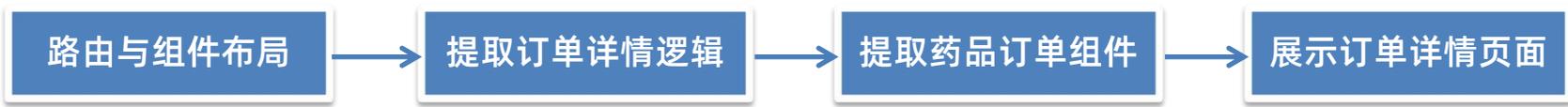
```
export const getMedicalOrderDetail = (id: string) =>  
  request<OrderDetail>(`/patient/medicine/order/detail/${id}`)
```

```
const order = ref<OrderDetail>()  
const route = useRoute()  
onMounted(async () => {  
  const res = await getMedicalOrderDetail(route.query.orderId as string)  
  order.value = res.data  
})  
  
<template>  
  <div class="order-pay-result-page">  
    <cp-nav-bar title="药品支付结果" />  
    <div class="result" v-if="order?.status === OrderType.ConsultPay">  
      <van-icon name="clear" />  
      <p class="price">¥ {{ order?.actualPayment }}</p>  
      <p class="status">支付失败</p>  
      <p class="tip">  
        订单支付失败，可以点击查看订单继续支付，如有疑问联系客服~  
      </p>  
    </div>  
    <div class="result" v-else>  
      <van-icon name="checked" />  
      <p class="price">¥ {{ order?.actualPayment }}</p>  
      <p class="status">支付成功</p>  
      <p class="tip">订单支付成功，已通知药房尽快发出，请耐心等待~</p>  
    </div>  
    <div class="btn">  
      <van-button type="primary" :to="'/order/${order?.id}'">  
        >查看订单</van-button  
      <van-button :to="'/room?orderId=${order?.roomId}'">返回诊室</van-button</div>  
  </div>  
</template>
```

05

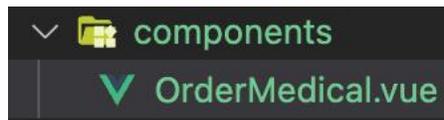
药品订单-订单详情

药品订单-订单详情



```
{  
  path: '/order/:id',  
  component: () => import('@/views/Order/OrderDetail.vue'),  
  meta: { title: '药品订单详情' }  
},
```

```
export const useOrderDetail = (id: string) => {  
  const loading = ref(false)  
  const order = ref<OrderDetail>()  
  onMounted(async () => {  
    loading.value = true  
    try {  
      const res = await getMedicalOrderDetail(id)  
      order.value = res.data  
    } finally {  
      loading.value = false  
    }  
  })  
  return { order, loading }  
}
```



```
const route = useRoute()  
const { order } = useOrderDetail(route.params.id as string)
```

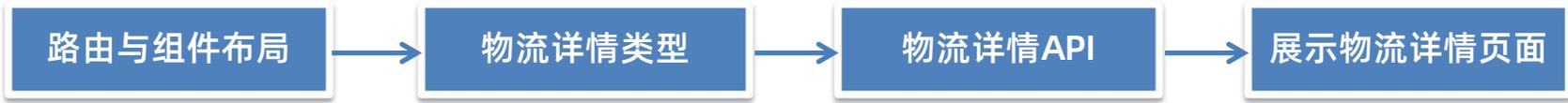
```
<cp-nav-bar :title="'药品订单-' + order.statusValue" />  
<div class="order-head">  
  <div  
    class="address" ←  
    v-if="  
      order.status === OrderType.MedicineCancel ||  
      order.status === OrderType.MedicinePay ||  
      order.status === OrderType.MedicineSend  
    "  
  >  
  <p class="area"> --  
  </p>  
  <p class="detail">{{ order.addressInfo.addressDetail }}</p>  
  <p> --  
  </p>  
</div>  
<div  
  class="card" ←  
  v-else  
  @click="$router.push('/order/logistics/${order?.id}')"  
>  
  <div class="logistics"> --  
  </div>  
  <van-icon name="arrow" />  
</div>  
</div>
```

```
<template  
  v-if="  
    order.status === OrderType.MedicineSend ||  
    order.status === OrderType.MedicineTake ||  
    order.status === OrderType.MedicineComplete  
  "  
>  
  <van-cell title="支付时间" :value="order.payTime" />  
  <van-cell  
    title="支付方式"  
    :value="order.paymentMethod === 0 ? '微信' : '支付宝'" />  
</template>
```

06

药品订单-物流详情

药品订单-物流详情



运输中 预计2023-03-22送达
申通e递 k20230320115431561999999

物流详情

- 运输中
快件到达【上海转运中心A1】
2023-03-21 02:54:32
- 运输中
快件到达【上海徐汇网点】
2023-03-21 02:54:32
- 已揽件
【上海徐汇网点】的徐快递 (13200000000) 已取件, 投诉电话: 020-10022123
2023-03-20 21:54:32
- 已发货
卖家已发货
2023-03-20 16:54:32
- 已发货
商品已下单,您的订单开始处理
2023-03-20 16:54:32

```
{  
  path: '/order/logistics/:id',  
  component: () => import('@views/Order/OrderLogistics.vue'),  
  meta: { title: '物流详情' }  
},
```

```
// 获取药品订单物流信息  
export const getMedicalOrderLogistics = (id: string) =>  
  request<Logistics>(`/patient/order/${id}/logistics`)
```

```
export type Express = {  
  /** 物流信息ID */  
  id: string  
  /** 物流内容 */  
  content: string  
  /** 创建时间 */  
  createTime: string  
  /** 物流状态 */  
  status: ExpressStatus  
  /** 状态文章 */  
  statusValue: string  
}  
  
export type Location = {  
  /** 经度 */  
  longitude: string  
  /** 纬度 */  
  latitude: string  
}  
  
export type Logistics = {  
  /** 预计送达时间 */  
  estimatedTime: string  
  /** 物流公司名称 */  
  name: string  
  /** 物流编号 */  
  awbNo: string  
  /** 最新物流状态 */  
  status: ExpressStatus  
  /** 最新物流状态文字 */  
  statusValue: string  
  /** 物流信息数组 */  
  list: Express[]  
  /** 轨迹信息数组 */  
  logisticsInfo: Location[]  
  /** 当前运输位置 */  
  currentLocationInfo: Location  
}
```

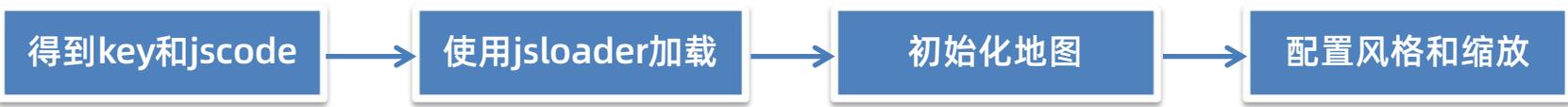
```
// 获取物流信息  
const logistics = ref<Logistics>()  
const route = useRoute()  
onMounted(async () => {  
  const res = await getMedicalOrderLogistics(route.params.id as string)  
  logistics.value = res.data  
})
```

```
<div id="map">  
  <div class="title">  
    <van-icon name="arrow-left" @click="$router.back()" />  
    <span>{{ logistics?.statusValue }}</span>  
    <van-icon name="service" />  
  </div>  
  <div class="current">  
    <p class="status">  
      {{ logistics?.statusValue }} 预计{{ logistics?.estimatedTime }}送达  
    </p>  
    <p class="predict">  
      <span>{{ logistics?.name }}</span>  
      <span>{{ logistics?.awbNo }}</span>  
    </p>  
  </div>  
</div>  
<div class="logistics">  
  <p class="title">物流详情</p>  
  <van-steps direction="vertical" :active="0">  
    <van-step v-for="item in logistics?.list" :key="item.id">  
      <p class="status" v-if="item.statusValue">{{ item.statusValue }}</p>  
      <p class="content">{{ item.content }}</p>  
      <p class="time">{{ item.createTime }}</p>  
    </van-step>  
  </van-steps>  
</div>
```

07

药品订单-使用高德地图

药品订单-使用高德地图



高德地图准备

```
onMounted(async () => {
  const res = await getMedicalOrderLogistics(route.params
  logistics.value = res.data

  initMap()
})

// 开发环境：配置JSCODE
window._AMapSecurityConfig = {
  securityJsCode: '415e917da833efcf2d5b69f4d821784b'
}

const initMap = async () => {
  // 加载高德地图
  const AMap = await AMapLoader.load({
    key: '4eed3d61125c8b9c168fc22414aaef7e',
    version: '2.0'
  })
}
```

```
// 初始化地图
const map = new AMap.Map('map', {
  mapStyle: 'amap://styles/whitesmoke',
  zoom: 12
})
```

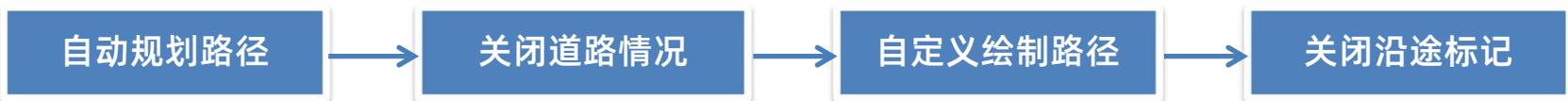
- 首先，注册开发者账号，成为高德开放平台开发者
- 登录之后，在进入「应用管理」页面「创建新应用」
- 为应用添加 Key，「服务平台」一项请选择「Web 端（JS API）」，设置域名白名单，可选（建议设置域名白名单）。

- 添加成功后，可获得key值和安全密钥jcode（自2021年12月02日升级，升级之后所申请的 key 必须配备安全密钥 jcode 一起使用）

08

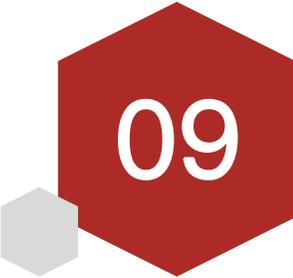
药品订单-自定义绘制路径

药品订单-自定义绘制路径



```
// 绘制轨迹
AMap.plugin('AMap.Driving', function () {
  const driving = new AMap.Driving({
    map,
    showTraffic: false,      2. 关闭道路情况
    hideMarkers: true       4. 隐藏沿途标记
  })
  if (
    logistics.value?.logisticsInfo &&
    logistics.value?.logisticsInfo.length >= 2
  ) {
    // 需要克隆不然影响页面
    const list = [...logistics.value.logisticsInfo]
    // 起点
    const start = list.shift()
    // 终点
    const end = list.pop()
    driving.search(
      [start?.longitude, start?.latitude], 1. 自动规划路径
      [end?.longitude, end?.latitude],
      { waypoints: list.map((item) => [item.longitude, item.latitude]) },
      () => {
        // 当前位置
      }
    )
  }
})
```





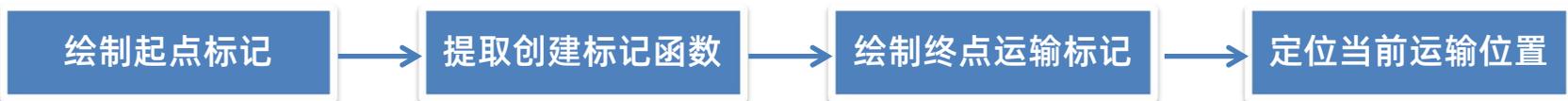
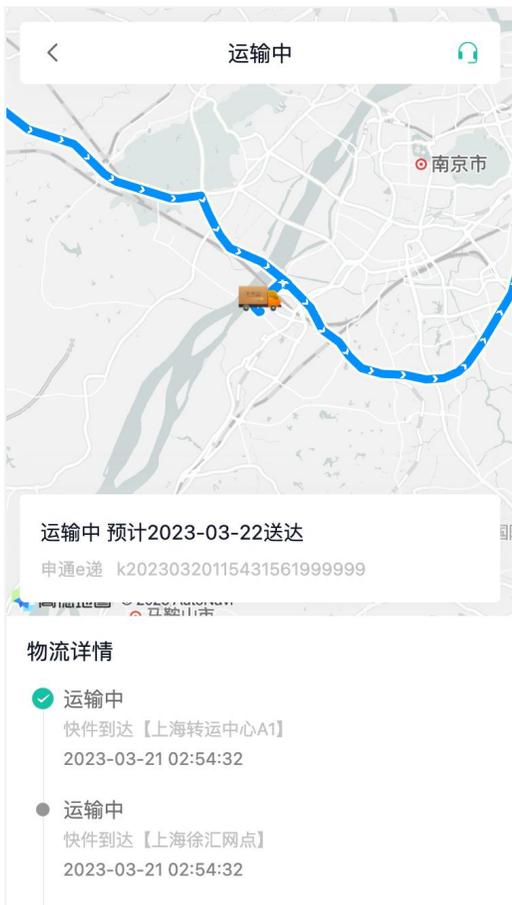
09

09-药品订单-绘制起点终点当前运输标记

10

10-药品订单-实现药品列表

药品订单-绘制起点终点当前运输标记



```
const getMarker = (  
  point: Location,  
  image: string,  
  width = 25,  
  height = 30  
  // 获取标记对象函数  
) => {  
  const icon = new AMap.Icon({  
    image,  
    size: new AMap.Size(width, height),  
    imageSize: new AMap.Size(width, height)  
  })  
  const Marker = new AMap.Marker({  
    position: [point.longitude, point.latitude],  
    icon: icon,  
    offset: new AMap.Pixel(-width / 2, -height)  
  })  
  return Marker  
}  
// 起点  
const start = list.shift()  
const startMarker = getMarker(start!, startImg)  
map.add(startMarker) // 添加起点  
// 终点  
const end = list.pop()  
const endMarker = getMarker(end!, endImg)  
map.add(endMarker) // 添加终点
```

```
driving.search(  
  [start?.longitude, start?.latitude],  
  [end?.longitude, end?.latitude],  
  { waypoints: list.map((item) => [item.longitude, item.latitude]) },  
  () => {  
    // 当前位置  
    const curr = logistics.value?.currentLocationInfo  
    const currMarker = getMarker(curr!, carImg, 33, 20)  
    map.add(currMarker)  
    // 当前运输位置  
    setTimeout(() => {  
      map.setFitView([currMarker])  
      map.setZoom(10)  
    }, 3000)  
  })  
}
```



传智教育旗下高端IT教育品牌