

云消息服务

# 开发指南

文档版本 01  
发布日期 2022-12-09



版权所有 © 华为技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 目录

<b>1 开发概述</b>	<b>1</b>
<b>2 总体开发思路</b>	<b>2</b>
<b>3 父组件开发</b>	<b>4</b>
3.1 编写 app-preview 组件的 JavaScript 逻辑	4
3.2 父组件代码示例	5
<b>4 子组件开发</b>	<b>6</b>
4.1 支持的子组件类型	6
4.2 事件处理效果预览	7
4.3 子组件代码示例	8
4.3.1 app-preview-text 组件	8
4.3.2 app-preview-button 组件	9
4.3.3 app-preview-image 组件	9
4.3.4 app-preview-video 组件	10
4.3.5 app-preview-sideslip 组件	10
4.3.6 app-preview-swiper 组件	13
4.3.7 app-preview-eCommerce 组件	14
4.3.8 app-preview-card-voucher 组件	16
4.3.9 app-preview-card-vouchers 组件	19
4.3.10 app-preview-moreGoods 组件	21
4.3.11 app-preview-ticketImage 组件	25
<b>5 修订记录</b>	<b>28</b>

# 1 开发概述

KooMessage是华为云为行业客户提供的一站式富媒体消息服务，聚合智能信息、5G消息、企业服务号、PUSH消息等多渠道触点，实现全场景、全终端触达。其中，智能信息通过手机增强技术提供可交互的富媒体内容和素材，可实现内容访问、信息查询、智能客服等与用户的实时互动，目前已支持十余类模板，提供上百个公共模板，适配多样化营销场景。

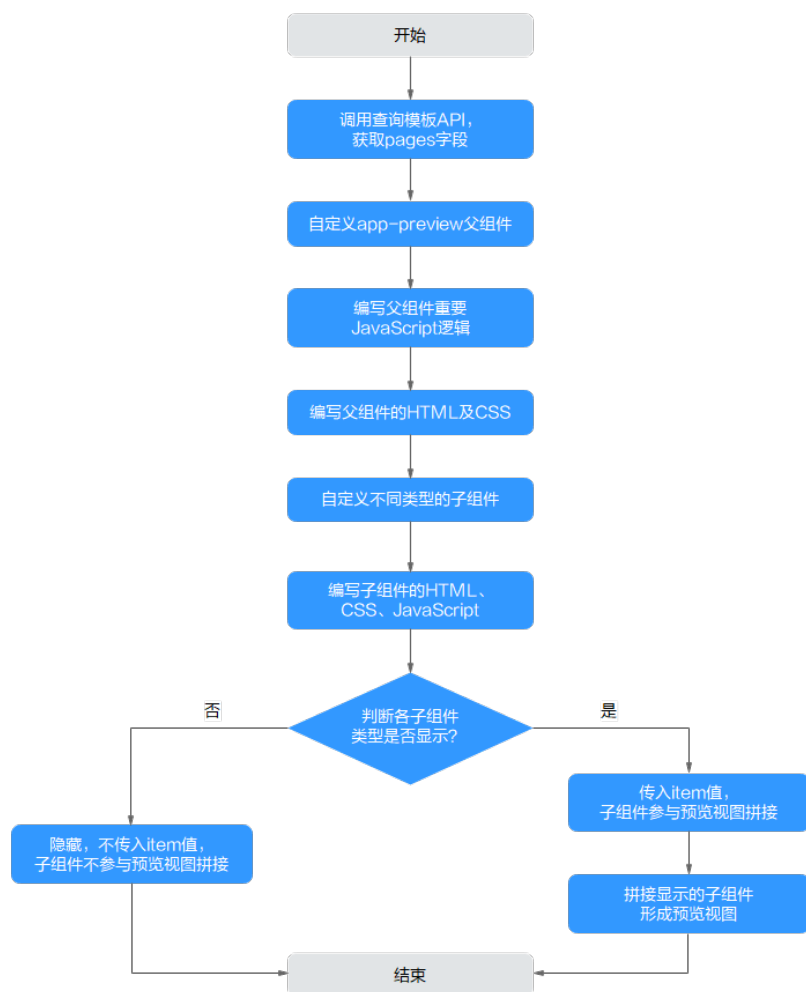
如果您是具备自用营销信息分发平台的行业客户，可选择将智能信息的模板数据接入自用平台，在自用平台上实现模板预览、模板管理等操作。为此您需要将模板查询API返回的JSON数据渲染成适配于自用平台的预览视图。

本指南将指导您如何将智能信息的模板数据渲染成前端预览视图。

# 2 总体开发思路

渲染模板预览视图所需的数据保存在pages字段中。调用智能信息模板查询API接口获取pages字段，经多个前端组件开发后，即可渲染出适配于自用平台的预览视图。总体开发流程如图2-1所示。

图 2-1 将模板数据渲染成预览视图的开发流程



## 关键阶段介绍

1. 调用查询模板API，获取pages字段。  
调用智能信息查询模板API，获取pages字段，具体调用方式请参见[如何调用API和智能信息查询模板API](#)。
2. 自定义app-preview父组件。  
app-preview父组件是一个包容所有子组件的容器，在app-preview父组件中定义一个类名为previews的div作为模板容器。
3. 编写父组件重要JavaScript逻辑。

### 📖 说明

- pages字段是字符串类型的参数，需先转换成JSON对象，便于后续传参。
- 缩放比例计算，确保模板显示比例正常，详情请参见[outerWidth](#)和[pages\[0\].style](#)。
  - 控制组件的大小和位置，详情请参见[pages\[0\].content](#)。
4. 编写父组件的HTML和CSS。  
在app-preview父组件中，定义一个类名为previews的div作为模板容器。再定义一个div容器，用来容纳遍历content数组获取的每个item对象。具体代码请参见[父组件代码示例](#)。
  5. 自定义不同类型的子组件。  
提供多种类型的子组件，如文本组件、图片组件、按钮组件等，可按需调用。
  6. 编写子组件的HTML、CSS、JavaScript。  
[子组件代码示例](#)提供了各类型子组件的HTML、CSS和JavaScript代码段供参考，开发时请以实际使用框架和组件库为准。
  7. 判断各子组件类型是否显示。  
遍历page[0].content获取item，根据item.type组件类型判断该子组件是否显示于目标可视界面中。如判断显示，则将item传入对应组件；如判断隐藏，则不传入。
  8. 拼接显示的子组件形成预览视图。  
判断显示的多个子组件拼接成预览视图，最终呈现在前端的预览视图即是包含了多个子组件的app-preview父组件。

# 3 父组件开发

## 3.1 编写 app-preview 组件的 JavaScript 逻辑

您可通过编写app-preview组件的重要js逻辑来设置原模板在预览界面的缩放比例、大小和位置，相关参数及用途如表3-1所示。

表 3-1 js 逻辑参数及用途

参数名	用途
<code>outerWidth</code>	控制模板在预览视图中的缩放尺寸。
<code>pages[0].style</code>	设置模板显示的缩放比例值。
<code>pages[0].content</code>	控制父组件的大小和位置。

### outerWidth

outerWidth参数用于控制模板在预览视图中的缩放尺寸，该参数应和包裹模板的div盒子在宽度上保持一致。假设盒子宽度为230px，则app-preview父组件的outerWidth字段值也应为230px。您可根据模板的实际展示需求自行决定outerWidth值，判断是否传入。

```
<div >  
// 将pages传入app-preview自定义组件，使用pages字段接收  
<app-preview [pages]="pages" [outerWidth]="230"></app-preview>  
</div>
```

### pages[0].style

在app-preview父组件中，用outerWidth除以pages[0].style.width得到一个缩放比例scale，然后将scale值赋给pages[0].style，以确保模板的整体显示比例正常。

```
let scale = outerWidth / parseInt(pages[0].style.width, 10);  
pages[0].style.transform = `scale(${scale})`
```

## pages[0].content

在app-preview父组件中，遍历content数组，为数组里每个item的style样式属性增加left、top、width、height属性，对应item.x、item.y、item.w、item.h，最后统一加上单位px，以控制每个子组件的大小和定位位置。

```
for (let item of pages[0].content) {  
  item.style = {  
    ...item.style,  
    left: joinPX(item.x),  
    top: joinPX(item.y),  
    width: joinPX(item.w),  
    height: joinPX(item.h)  
  };  
}  
joinPX(data) {  
  return data + 'px';  
}
```

### 📖 说明

目前在pages的数据对象中，除pages[0].style和pages[0].content外的参数暂未使用。

## 3.2 父组件代码示例

### HTML

```
<div class="previews" [style]=" pages[0].style" >  
  // ngFor为循环指令，遍历content数组获取item  
  <div *ngFor="let item of pages[0].content">  
    // ngIf为判断指令，true为显示，false为隐藏  
    // [item]=" item" 是将item数据传入对应组件，组件内用item变量接收  
    <app-preview-text *ngIf="item.type === 'text'" [item]="item"></app-preview-text>  
    <app-preview-button *ngIf="item.type === 'button'" [item]="item" ></app-preview-  
button>  
    <app-preview-image *ngIf="item.type === 'image'" [item]="item"></app-preview-  
image>  
    <app-preview-video *ngIf="item.type === 'video'" [item]="item"></app-preview-video>  
    <app-preview-sideslip *ngIf="item.type === 'sideslip'" [item]="item"></app-preview-  
sideslip>  
    <app-preview-swiper *ngIf="item.type === 'swiper'" [item]="item"></app-preview-  
swiper>  
    <app-preview-ecommerce *ngIf="item.type === 'eCommerce'" [item]="item"></app-preview-  
ecommerce>  
    <app-preview-card-voucher *ngIf="item.type === 'card-voucher'" [item]="item"></app-  
preview-card-voucher>  
    <app-preview-card-vouchers *ngIf="item.type === 'card-vouchers'" [item]="item"></app-  
preview-card-vouchers>  
  </div>  
</div>
```

### CSS

```
.previews {  
  position: relative;  
  transform-origin: left top;  
}
```



# 4 子组件开发

## 4.1 支持的子组件类型

以图4-1为例，该模板的预览视图由image、text、button三个子组件拼接而成。目前KooMessage的智能短信模板已支持11种类型的子组件开发，详见表4-1。对于各类型子组件，[子组件代码示例](#)提供了相应的HTML、CSS、JavaScript代码段供参考，部分组件无需编写CSS样式和JavaScript逻辑，请您以实际使用框架和组件库为准。

图 4-1 渲染出的模板预览视图



表 4-1 支持的组件类型和判断条件

组件类型	对应的判断条件	对应的自定义组件
文本	<code>item.type === 'text'</code>	<code>app-preview-text</code>
按钮	<code>item.type === 'button'</code>	<code>app-preview-button</code>

组件类型	对应的判断条件	对应的自定义组件
图片	item.type === 'image'	app-preview-image
视频	item.type === 'video'	app-preview-video
横滑	item.type === 'sideslip'	app-preview-sideslip
轮播	item.type === 'swiper'	app-preview-swiper
电商	item.type === 'eCommerce'	app-preview-ecommerce
卡券1	item.type === 'cardVoucher'	app-preview-card-voucher
卡券2	item.type === 'cardVouchers'	app-preview-card-vouchers
多商品	item.type === 'moreGoods'	app-preview-moreGoods
增强机票	item.type === 'ticketImage'	app-preview-ticketImage

## 4.2 事件处理效果预览

“事件”是指用户在客户端查看智能短信时，通过与子组件交互可完成的客户端操作。比如手机用户点击短信中的“进入官网”按钮，即可访问官网。

如您需要在预览模板时查询模板绑定的事件处理效果，这一需求可通过调用 PreviewService 函数统一处理事件来实现。目前 item 元素支持表 4-2 中列举的 12 种事件类型，您可根据实际情况给组件赋予事件类型。

### 说明

目前 PreviewService 函数仅可处理 OPEN\_APP（仅支持移动端，在非移动端操作会有弹窗提示）、OPEN\_BROWSER、OPEN\_URL 事件，VIEW\_PIC 事件需在 app-preview-image 中单独处理。

## 模板预览事件类型

表 4-2 事件类型及 ActionTypes 枚举值

事件类型	ActionTypes 枚举值
查看大图	VIEW_PIC = 'VIEW_PIC'
打开 URL	OPEN_URL = 'OPEN_URL'
打开浏览器	OPEN_BROWSER = 'OPEN_BROWSER'
复制参数	COPY_PARAMETER = 'COPY_PARAMETER'
打开 APP	OPEN_APP = 'OPEN_APP'
打开地图	OPEN_MAP = 'OPEN_MAP'
拨打电话	DIAL_PHONE = 'DIAL_PHONE'

事件类型	ActionTypes枚举值
打开快应用	OPEN_QUICK = 'OPEN_QUICK'
打开短信	OPEN_SMS = 'OPEN_SMS'
打开日程表	OPEN_SCHEDULE = 'OPEN_SCHEDULE'
打开邮箱	OPEN_EMAIL = 'OPEN_EMAIL'
打开弹窗	OPEN_POPUP = 'OPEN_POPUP'

## PreviewService 函数调用示例

```
import { ActionTypes };
export PreviewService {
  emit($event) {
    const { action, target } = $event;
    switch (action) {
      case ActionTypes.OPEN_APP:
        if (target) {
          this.alert(`你要打开的app地址为${target},请在移动端进行此操作`);
        }
        break;
      case ActionTypes.OPEN_BROWSER:
      case ActionTypes.OPEN_URL:
        location.href = target;
        break;
      default:
        break;
    }
  }
}
```

## 4.3 子组件代码示例

### 4.3.1 app-preview-text 组件

#### HTML

```
<div>
  <p
    class="preview-element text-element"
    [style]="item.style"
    (click)="PreviewService.emit(item.event)"
    // innerHTML为插入html片段的指令
    [innerHTML]="item.text"
  ></p>
</div>
})
```

#### CSS

```
.text-element {
  overflow: hidden;
  overflow-y: auto;
```

```
white-space: pre-wrap;
input {
  color: inherit;
  border: 0;
  outline: none;
  background: hsla(0, 0%, 100%, 0);
  padding: 0;
  font-size: inherit;
}
input[type='button'],
input[type='reset'],
input[type='search'],
input[type='submit'] {
  -webkit-appearance: none;
  cursor: pointer;
}
}
```

## JavaScript

```
// 引入PreviewService, 用来触发事件
import { PreviewService };
```

### 4.3.2 app-preview-button 组件

#### HTML

```
<div>
<p [style]="item.style" (click)="PreviewService.emit(item.event)" class="preview-element" >
  {{ item.text }}
</p>
</div>
```

#### CSS

```
.preview-element {
  display: flex;
  justify-content: center;
  align-items: center;
  &.XiaoMi {
    color: rgb(0, 0, 0) !important;
    background-color: rgb(255, 255, 255) !important;
  }
  &.VIVO {
    color: rgb(59, 158, 255) !important;
    background-color: initial !important;
  }
}
```

## JavaScript

```
// 引入PreviewService, 用来触发事件
import { PreviewService };
```

### 4.3.3 app-preview-image 组件

#### HTML

```
<div class="imgBox preview-element">
<img
```

```
class="preview-element"
[src]="item?.src"
[style]="item.style"
// 若图片的动作类型为查看大图，这里使用viewer库实现，开发者请根据实际情况编码
[view]="item?.event?.action === ActionTypes.VIEW_PIC"
(click)="PreviewService.emit(item.event)"
/>
</div>
})
```

## CSS

无。

## JavaScript

```
import { ActionTypes }
import { PreviewService }
// 使用Viewer这个图片查看相关的js库
import Viewer from 'viewerjs';
```

### 4.3.4 app-preview-video 组件

#### HTML

```
<div>
<video class="preview-element" [poster]="item.cover" controls [style]="item.style"
[src]="item.src"></video>
</div>
```

## CSS

无。

## JavaScript

无。

### 4.3.5 app-preview-sideslip 组件

#### HTML

```
<div class="Sideslip" [style]="item.style">
<div class="Sideslip-scroll" [style]="{ width: sideslipList.length * 210 + 16 * (sideslipList.length -
1) + 'px' }">
<div
// ngFor为循环指令
*ngFor="let list of sideslipList; index as i"
class="Sideslip-box"
(click)="changeIndex(i)"
[class]="{
active: i === activeIndex
}"
>
<div class="Sideslip-file">
<img
[src]="list.src"
alt=""
```

```
[style]="{ height: (208 * proportionM) / proportionD + 'px' }"  
(click)="handlerAction(i)"  
</div>  
</div>  
<div class="Sideslip-title" [innerHTML]="list.title"></div>  
<div *ngIf="item.contentShow" class="Sideslip-cotent" [innerHTML]="list.content |"></div>  
<div *ngIf="list.buts.length > 0" class="Sideslip-buts" (click)="handlerAction(i)">  
<div *ngFor="let item1 of list.buts" class="Sideslip-but">  
  {{ item1.text }}  
</div>  
</div>  
</div>  
</div>  
</div>  
</div>
```

## CSS

```
Sideslip {  
  position: relative;  
  overflow-x: auto;  
  overflow-y: hidden;  
  .Sideslip-scroll {  
    position: absolute;  
    left: 0;  
    top: 0;  
  }  
  .Sideslip-box {  
    display: inline-block;  
    margin-right: 16px;  
    width: 210px;  
    border: 1px solid #c9c9c9;  
    border-radius: 8px;  
    overflow: hidden;  
    background-color: #fff;  
    box-sizing: border-box;  
    cursor: pointer;  
    &.active {  
      border-color: #5e7ce0;  
    }  
    &:last-of-type {  
      margin-right: 0;  
    }  
  }  
  .Sideslip-file {  
    img {  
      width: 208px;  
    }  
  }  
  .Sideslip-title {  
    color: #333;  
    min-height: 20px;  
    line-height: 20px;  
    font-size: 15px;  
    margin-left: 12px;  
    margin-bottom: 8px;  
    margin-top: 8px;  
    padding-right: 4px;  
    box-sizing: border-box;  
    font-weight: 700;  
    text-overflow: ellipsis;  
    white-space: nowrap;  
    overflow: hidden;  
  }  
  .Sideslip-cotent {
```

```
color: #666;
height: 36px;
line-height: 18px;
font-size: 12px;
margin-left: 12px;
margin-bottom: 14px;
display: -webkit-box;
-webkit-box-orient: vertical;
-webkit-line-clamp: 2;
overflow: hidden;
}
.Sideslip-buts {
width: 100%;
padding-bottom: 15px;
box-sizing: border-box;
.Sideslip-but {
text-align: center;
width: 80%;
margin: 0 auto;
height: 40px;
line-height: 40px;
border-radius: 20px;
background-color: #007dff;
color: #fff;
font-size: 17px;
}
}
}
}
}
::-webkit-scrollbar {
width: 8px;
height: 8px;
}
::-webkit-scrollbar-track-piece {
background-color: none;
background: rgba(0, 0, 0, 0.1);
}
::-webkit-scrollbar-thumb:vertical {
height: 8px;
background: rgba(0, 0, 0, 0.5);
}
::-webkit-scrollbar-thumb:horizontal {
width: 8px;
background: rgba(0, 0, 0, 0.5);
}
```

## JavaScript

```
import { PreviewService }
// 自定义的一些参数
sideslipList = [];
proportionM = 1;
proportionD = 1;
activeIndex = 0;
sideboxElements;
// 初始化数据
ngOnInit() {
this.sideslipList = this.item?.sideslip?.sideslipList || [];
this.activeIndex = this.item?.sideslip?.sideslipList || 0;
// 获取dom节点
this.sideboxElements = this..getElementsByClassName('Sideslip-box');
```

```
}  
// 切换事件  
changeIndex(index) {  
  let element = this.sideboxElements[index];  
  element.scrollIntoView({ inline: 'start' });  
  this.activeIndex = index;  
}  
handlerAction(index) {  
  let current = this.sideslipList[index];  
  this.previewService.emit(current.buts[0].event);  
}  
}
```

### 4.3.6 app-preview-swiper 组件

开发横滑组件时可定义的属性及其作用如表4-3所示，您可根据实际需求进行定义。

表 4-3 横滑组件属性及作用

属性	作用
item.style	样式定义。
item.swiper.autoplay	是否自动轮播。
item.swiper.loop	是否循环。
item.swiper.showIndicatorPoint	是否显示底部指示器。

#### HTML

```
<div>  
<swiper  
  [style]="item.style"  
  [autoplay]="item.swiper.autoplay"  
  [loop]="item.swiper.loop"  
  [indicatorPosition]="item.swiper.showIndicatorPoint ? 'bottom' : 'none'  
>  
  // ngFor为循环指令  
  <swiperSlide*ngFor="let item1 of item.swiper.swiperArr">  
    <img [style]="{ width: item.style.width }" [src]="item1.src" alt="" />  
  </swiperSlide >  
</swiper>  
</div>
```

#### CSS

无。

#### JavaScript

无。



## 4.3.7 app-preview-eCommerce 组件

### HTML

```
<div class="ECommerce">
  <swiper [autoplay]="false" [loop]="true" *ngIf="item.content.imageList.visibility === 'visible'">
    // ngFor为循环指令
    <swiperSlide *ngFor="let item of item.content.imageList.listItem">
      <div class="swiperImg">
        <img [src]="item.image.src" />
      </div>
    </swiperSlide >
  </swiper>
  <video
    controls
    *ngIf="item.content.video.visibility === 'visible'"
    [src]="item.content.video.src"
    [poster]="item.content.video.cover"
  ></video>
  <div class="price">
    <span
      class="price_symbol"
      *ngIf="
        item.content.divTextArea1.currencyText ? item.content.divTextArea1.currencyText.visibility ===
        'visible' : true
      "
    >
    </span>
    <span class="price_real">{{ item.content.divTextArea1.text1.textValue }}</span>
    <span class="price_favorable">{{ item.content.divTextArea1.text2.textValue }}</span>
  </div>
  <div class="divTitle">
    {{ item.content.divTextArea2.textTitle.textValue }}
  </div>
  <div class="divContent">
    {{ item.content.divTextArea2.textContent.textValue }}
  </div>
  <div class="divButton">
    <div [class]="{ divButton_left: true, oneButton: item.content.divButton.length === 1 }">
      {{ item.content.divButton[0].textValue }}
    </div>
    <div class="divButton_right" *ngIf="item.content.divButton[1]">
      {{ item.content.divButton[1].textValue }}
    </div>
  </div>
</div>
```

### CSS

```
.ECommerce .divButton {
  display: flex;
  justify-content: space-around;
}
.ECommerce .divButton .oneButton {
  width: 280px;
  background-color: #e3413a;
}
.ECommerce .divButton_left,
.ECommerce .divButton_right {
```

```
width: 140px;
height: 36px;
color: #fff;
text-align: center;
line-height: 36px;
font-size: 16px;
border-radius: 40px;
}
.ECommerce .divButton_left {
background-color: #ef9c38;
}
.ECommerce .divButton_right {
background-color: #e3413a;
}
.ECommerce .divContent {
font-size: 13px;
display: -webkit-box;
-webkit-box-orient: vertical;
-webkit-line-clamp: 3;
margin-bottom: 20px;
height: 50px;
}
.ECommerce .divContent,
.ECommerce .divTitle {
padding-left: 24px;
padding-right: 24px;
margin-top: 8px;
margin-bottom: 8px;
overflow: hidden;
}
.ECommerce .divTitle {
font-weight: 700;
font-size: 16px;
text-overflow: ellipsis;
white-space: nowrap;
}
.ECommerce .price {
color: #e3413a;
margin-top: 20px;
padding-left: 24px;
box-sizing: border-box;
white-space: nowrap;
}
.ECommerce .price .price_symbol {
font-size: 14px;
}
.ECommerce .price .price_real {
font-size: 20px;
font-weight: 700;
}
.ECommerce .price .price_favorable {
background-color: #e3413a;
color: #fff;
border-radius: 2px;
display: inline-block;
padding: 1px 4px;
box-sizing: border-box;
transform: translateY(3px);
margin-left: 4px;
font-size: 14px;
letter-spacing: 1px;
max-width: 150px;
```

```
overflow: hidden;
text-overflow: ellipsis;
white-space: nowrap;
}
.ECommerce video {
width: 342px;
height: 336px;
}
.swiperImg {
img {
width: 342px;
height: 342px;
display: block;
margin: 0 auto;
}
}
```

## JavaScript

无。

## 4.3.8 app-preview-card-voucher 组件

### HTML

```
<div class="card-voucher">
<div class="content">
<div class="content_top">
<div *ngIf="item.content.voucheType === 0" class="content_top_text1">¥</div>
<div class="content_top_text2">
{{headLeft}}
</div>
<div class="content_top_textBox">
<p class="content_top_text3">{{ headRightTop }}</p>
<p class="content_top_text4">
{{headRightBottom}}
</p>
</div>
</div>
<div class="content_middle" [innerHTML]="item.content.divText.text5.textValue"></div>
<div class="content_bottom">
{{ transform (item)}}
</div>
</div>
<img class="logo" (click)="events(item.content.divImage)" [src]="item.content.divImage.src"
alt="" />
<div (click)="events(item.content.divButton)"
class="button">{{ item.content.divButton.textValue }}</div>
</div>
```

### CSS

```
.card-voucher {
position: relative;
height: 364px;
background-image: url(https://aim6.oss-cn-beijing.aliyuncs.com/editor_file/card-voucherBg.png);
background-size: cover;
background-position: center;
background-repeat: no-repeat;
.logo {
```

```
width: 70px;
height: 70px;
position: absolute;
left: 50%;
margin-left: -35px;
top: 180px;
bottom: 10px;
right: 20px;
border-radius: 70px;
}
.button {
width: 290px;
height: 40px;
background-color: #ffbf00;
border-radius: 20px;
text-align: center;
color: #fff;
line-height: 40px;
font-size: 16px;
position: absolute;
bottom: 35px;
left: 26px;
}
.content {
padding-top: 50px;
box-sizing: border-box;
&_top {
display: flex;
justify-content: center;
color: #fa282b;
height: 60px;
align-items: center;
&_text1 {
font-size: 20px;
margin-right: 0px;
padding-top: 20px;
box-sizing: border-box;
}
&_text2 {
font-size: 46px;
margin-right: 10px;
font-weight: bold;
}
&_textBox {
&_content_top_text3 {
border: 1px solid #fa282b;
padding: 1px 3px;
border-radius: 4px;
box-sizing: border-box;
font-size: 12px;
margin-bottom: 2px;
display: inline-block;
}
&_content_top_text4 {
font-size: 17px;
}
}
}
&_middle {
text-align: center;
color: #a62929;
font-size: 13px;
```

```
font-weight: 700;
height: 18px;
width: 280px;
margin: 0 auto 8px;
word-break: break-all;
text-overflow: ellipsis;
overflow: hidden;
display: -webkit-box;
-webkit-line-clamp: 2;
-webkit-box-orient: vertical;
}
&_bottom {
text-align: center;
color: #a62929;
font-size: 12px;
}
}
}
```

## JavaScript

```
import { PreviewService };
// 自定义的参数
headLeft;
headRightTop;
headRightBottom;
// 初始化函数
ngOnInit(){
this.inintData();
}
inintData() {
let type = this.item.content.voucheType;
if (type === 0) {
this.headLeft = this.item.content.fullMinus.reduction;
this.headRightTop = '满减券';
this.headRightBottom = `满${this.item.content.fullMinus.money}可用`;
} else if (type === 1) {
this.headLeft = this.item.content.discount.fracture + '折';
this.headRightTop = '折扣券';
this.headRightBottom = `满${this.item.content.discount.money}可用`;
} else if (type === 2) {
this.headLeft = this.item.content.custom.money;
this.headRightTop = this.item.content.custom.tagName;
this.headRightBottom = this.item.content.custom.threshold;
}
}
// 点击事件
events(val) {
this.PreviewService.emit(val.event);
}
// 对数据做判断处理
transform(item, ...args ){
if (item.dateType === 0) {
return item.fixedDate.join('~');
} else if (item.dateType === 1) {
return `领取后${item1.termValidity}天后有效`;
} else if (item.content?.voucheType === 2) {
return `${item.content.fixedDate[0]}~${item.content.fixedDate[1]}`;
}
return null;
}
```

## 4.3.9 app-preview-card-vouchers 组件

### HTML

```
<div [style]="item.style" class="cardVouchers">
// innerHTML会把代码片段插入到当前节点中
<div class="title" [innerHTML]="item.content.divText.text1.textValue"></div>
<div class="content">
// ngFor为循环指令
<div class="content-item" *ngFor="let item1 of item.content.divItemList">
<div class="content-item-left">
<div class="content-item-left-tip">{{ item1.voucheType === 0 ? '满减卷' : '折扣卷' }}</div>
<div class="content-item-left-middle">
<span *ngIf="item1.voucheType === 0" class="symbol">¥</span>
<span class="money">{{
item1.voucheType === 0 ? item1.fullMinus.reduction : item1.discount.fracture + '折'
}}</span>
</div>
<div class="content-item-left-bottom">
满{{ item1.voucheType === 0 ? item1.fullMinus.money : item1.discount.money }}可用
</div>
</div>
<div class="content-item-right">
<div class="content-item-right-name" [innerHTML]="item1.divText.text5.textValue"></div>
<div class="content-item-right-date">
{{ transform(item1) }}
</div>
<div (click)="events(item1.divButton)" *ngIf="item.content.buttonType === 1" class="content-
item-right-button">
{{ item1.divButton.textValue }}
</div>
</div>
</div>
</div>
<div (click)="events(item.content.divButton)" *ngIf="item.content.buttonType === 0"
class="button">
{{ item.content.divButton.textValue }}
</div>
</div>
```

### CSS

```
.cardVouchers {
height: 100%;
background-color: #f32c3d;
.title {
font-size: 20px;
color: #fff;
text-align: center;
line-height: 64px;
background-color: initial;
overflow: hidden;
text-overflow: ellipsis;
white-space: nowrap;
padding: 0 12px;
box-sizing: border-box;
}
.button {
width: 290px;
height: 40px;
background-color: #ffbf00;
```

```
border-radius: 20px;
text-align: center;
color: #fff;
line-height: 40px;
font-size: 16px;
margin: 0 auto;
}
.content {
padding: 0 9px 8px;
box-sizing: border-box;
&-item {
height: 90px;
background-color: #fff5eb;
border-radius: 7px;
margin-bottom: 10px;
position: relative;
display: flex;
align-items: center;
&-left {
width: 104px;
height: 100%;
text-align: center;
padding-top: 20px;
box-sizing: border-box;
&-tip {
background-color: #feded6;
border-radius: 7px 0 7px 0;
padding: 3px 5px;
position: absolute;
left: 0;
top: 0;
color: #fa282b;
font-size: 12px;
}
&-middle {
color: #fa282b;
.symbol {
font-size: 14px;
margin-right: 2px;
}
}
.money {
font-size: 30px;
}
}
&-bottom {
font-size: 12px;
color: #a62929;
}
}
&-right {
border-left: 2px solid #febab4;
height: 72px;
flex: 1;
padding: 0 8px;
box-sizing: border-box;
&-name {
color: #a62929;
font-size: 13px;
font-weight: 700;
height: 40px;
line-height: 20px;
margin-bottom: 17px;
```

```
word-break: break-all;
text-overflow: ellipsis;
overflow: hidden;
display: -webkit-box;
-webkit-line-clamp: 2;
-webkit-box-orient: vertical;
text-align: left;
}
&-date {
color: #a62929;
font-size: 12px;
text-align: left;
}
&-button {
width: 58px;
height: 26px;
border-radius: 27px;
background-color: #ffbf00;
color: #fff;
text-align: center;
line-height: 26px;
font-size: 12px;
position: absolute;
bottom: 4px;
right: 4px;
}
}
}
}
```

## JavaScript

```
// 点击事件
events(val) {
this.PreviewService.emit(val.event);
}
// 对数据做判断处理
transform(item, ...args ){
if (item.dateType === 0) {
return item.fixedDate.join('~');
} else if (item.dateType === 1) {
return `领取后${item1.termValidity}天后有效`;
} else if (item.content?.voucheType === 2) {
return `${item.content.fixedDate[0]}~${item.content.fixedDate[1]}`;
}
return null;
}
```

### 4.3.10 app-preview-moreGoods 组件

#### HTML

```
<div :style="item.style" class="moreGoods">
  <div class="title">
// innerHTML会把代码片段插入到当前节点中
    <span class="title-name" [innerHTML]="item.content.divheader.text1.textValue"></span>
    <span
      class="title-icon"
      (click)="events(item.content.divheader.imagelcon)"
      *ngIf="item.content.divheader.imagelcon.visibility === 'visible'"
```



```
>
  <ti-icon name="angle-right"></ti-icon>
</span>
</div>
</div>
<div class="subhead" [innerHTML]="item.content.divheader.text2.textValue"></div>
<div class="content">
  <div
    class="content-item"
    *ngFor="let item of item.content.divItemlist; index as index"
    (click)="events(item.divTextButton)"
  >
    <div class="content-item-imgBox">
      <img [src]="item.divTextImage.image1.src" alt="" />
    </div>
    <div class="content-item-good">
      <div class="content-item-good-name"
[innerHTML]="item.divTextButton.text4.textValue"></div>
      <div class="content-item-good-tags">
        <span [innerHTML]="item.divTextButton.text5.textValue"></span>
        <span
          *ngIf="item.divTextButton.text7.textValue"
          [innerHTML]="item.divTextButton.text7.textValue"
        ></span>
      </div>
      <div class="content-item-good-buttonBox">
        <div class="content-item-good-buttonBox-price">
          <span
            *ngIf="item.divTextButton.text.visibility === 'visible'"
            class="content-item-good-buttonBox-price-symbol"
            >{{ item.divTextButton.text.textValue }}</span>
          >
          <span
            [innerHTML]="item.divTextButton.text6.textValue"
            class="content-item-good-buttonBox-price-money"
          ></span>
        </div>
      </div>
      <div
        class="content-item-good-buttonBox-button"
        [innerHTML]="item.divTextButton.button1.textValue"
      ></div>
    </div>
  </div>
</div>
</div>
</div>
</div>
```

## CSS

```
.moreGoods {
  height: 100%;
  box-sizing: border-box;
  background-color: #ffdada;
  background-image: url(https://aim6.oss-cn-beijing.aliyuncs.com/editor_file/goodsBg.png);
  background-position: top;
  background-size: 100% auto;
  background-repeat: no-repeat;
  .title {
    font-size: 20px;
    color: #fff;
    text-align: center;
    line-height: 36px;
    height: 36px;
  }
}
```

```
background-color: initial;
overflow: hidden;
text-overflow: ellipsis;
white-space: nowrap;
padding: 15px 12px 0;
justify-content: center;
align-items: center;
display: flex;
&-name {
  max-width: 280px;
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
}
&-icon {
  color: #ff0d51;
  background-color: #fff;
  display: inline-block;
  border-radius: 20px;
  width: 30px;
  height: 20px;
  line-height: 20px;
  font-size: 14px;
  margin-left: 6px;
  ti-icon {
    font-weight: bold;
  }
}
}
.subhead {
  font-size: 14px;
  color: #fff;
  text-align: center;
  line-height: 24px;
  height: 24px;
  background-color: initial;
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
  box-sizing: border-box;
  margin-bottom: 12px;
}
.content {
  &-item {
    background-color: #fff;
    height: 122px;
    width: 324px;
    margin: 0 auto;
    border-radius: 8px;
    margin-bottom: 10px;
    padding: 8px;
    box-sizing: border-box;
    display: flex;
    &-imgBox {
      width: 106px;
      height: 106px;
      img {
        width: 100%;
        height: 100%;
        border-radius: 8px;
      }
    }
  }
}
```

```
&-good {
  flex: 1;
  margin-left: 12px;
  &-name {
    overflow: hidden;
    text-overflow: ellipsis;
    width: 100%;
    display: -webkit-box;
    -webkit-box-orient: vertical;
    -webkit-line-clamp: 2;
    word-break: break-all;
    line-height: 22px;
    margin-bottom: 4px;
    height: 44px;
  }
  &-tags {
    height: 20px;
    overflow-y: auto;
    margin-bottom: 8px;
    span {
      border: 1px solid #fa2a2d;
      color: #fa2a2d;
      border-radius: 3px;
      font-size: 12px;
      padding: 0px 3px;
      &:first-child {
        margin-right: 3px;
      }
    }
  }
}
&-buttonBox {
  display: flex;
  justify-content: space-between;
  line-height: 30px;
  &-price {
    font-size: 18px;
    color: #fa2a2d;
    display: flex;
    align-items: center;
    &-symbol {
      font-size: 15px;
    }
  }
  &-money {
    display: inline-block;
    max-width: 98px;
    text-overflow: ellipsis;
    overflow: hidden;
    white-space: nowrap;
  }
}
&-button {
  width: 80px;
  height: 30px;
  line-height: 30px;
  background-color: #fa2a2d;
  border-radius: 30px;
  text-align: center;
  color: #fff;
  font-size: 14px;
  text-overflow: ellipsis;
  overflow: hidden;
  white-space: nowrap;
}
```

```
    }  
  }  
}  
}
```

## JavaScript

```
// 点击事件  
events(val) {  
  this.PreviewService.emit(val.event);  
}
```

### 4.3.11 app-preview-ticketImage 组件

#### HTML

```
<div [style]="item.style" class="ticketImage">  
  <div class="ticketImage_List">  
    <div  
      // ngFor为循环指令  
      *ngFor="let listItem of item.content.divList1.list; index as i1"  
      (click)="events(listItem.img)"  
      class="ticketImage_List_item"  
      [ngClass]="{ ticketImage_List_item_two: item.content.divList1.list.length === 2 }"  
    >  
      <img [src]="listItem.img.src" alt="" srcset="" />  
      <span [innerHTML]="listItem.text.textValue"></span>  
    </div>  
  </div>  
  <div class="ticketImage_TextList">  
    <div *ngFor="let listItem2 of item.content.divList2.list; index as i2"  
      class="ticketImage_TextList_item">  
      // innerHTML会把代码片段插入到当前节点中  
      <div class="ticketImage_TextList_item_title" [innerHTML]="listItem2.textUp.textValue"></div>  
      <div class="ticketImage_TextList_item_content"  
        [innerHTML]="listItem2.textDown.textValue"></div>  
    </div>  
  </div>  
  <div class="ticketImage_longText">  
    <div *ngFor="let listItem3 of item.content.divList3.list; index as i3"  
      class="ticketImage_longText_item">  
      <div class="ticketImage_longText_item_title" [innerHTML]="listItem3.textUp.textValue |  
        html"></div>  
      <div class="ticketImage_longText_item_content"  
        [innerHTML]="listItem3.textDown.textValue"></div>  
    </div>  
  </div>  
  <div class="ticketImage_tip" [innerHTML]="item.content.divTip.text.textValue"></div>  
</div>
```

#### CSS

```
.ticketImage {  
  padding: 0 23px !important;  
  box-sizing: border-box;  
  position: absolute;  
  .ticketImage_List {  
    display: flex;
```

```
text-align: center;
justify-content: center;
margin-bottom: 18px;
&_item {
  width: 60px;
  margin-right: 15px;
  &:last-child {
    margin-right: 0;
  }
  img {
    width: 38px;
    height: 38px;
  }
  span {
    display: block;
    color: #666666;
    font-size: 12px;
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
  }
}
&_item_two {
  width: 120px;
  display: flex;
  align-items: center;
  margin-top: 10px;
  margin-bottom: 10px;
  img {
    margin-right: 8px;
  }
}
}
.ticketImage_TextList {
  display: flex;
  flex-wrap: wrap;
  margin-bottom: 6px;
  &_item {
    width: 33.33%;
    text-align: left;
    margin-bottom: 6px;
    &:nth-child(3n - 1) {
      text-align: center;
    }
    &:nth-child(3n) {
      text-align: right;
    }
  }
  &_title {
    color: #a9a9a9;
    font-size: 12px;
    margin-bottom: 2px;
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
    height: 16px;
  }
  &_content {
    color: #1d1d1d;
    font-size: 14px;
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
  }
}
```

```
        height: 19px;
      }
    }
  }
}
.ticketImage_longText {
  margin-bottom: 12px;
  &_item {
    margin-bottom: 6px;
    &_title {
      color: #a9a9a9;
      font-size: 12px;
      margin-bottom: 2px;
      overflow: hidden;
      text-overflow: ellipsis;
      height: 16px;
      white-space: nowrap;
    }
    &_content {
      color: #1d1d1d;
      font-size: 14px;
      overflow: hidden;
      text-overflow: ellipsis;
      white-space: nowrap;
      height: 19px;
    }
  }
}
.ticketImage_tip {
  height: 42px;
  overflow-y: auto;
  color: #a9a9a9;
  font-size: 14px;
  &::-webkit-scrollbar {
    width: 4px;
  }

  &::-webkit-scrollbar-track {
    background-color: #fff;
    -webkit-border-radius: 2em;
    -moz-border-radius: 2em;
    border-radius: 2em;
  }

  &::-webkit-scrollbar-thumb {
    background-color: #e6d9d9;
    -webkit-border-radius: 2em;
    -moz-border-radius: 2em;
    border-radius: 2em;
  }
}
}
```

## JavaScript

```
// 点击事件
events(val) {
  this.PreviewService.emit(val.event);
}
```

# 5 修订记录

---

发布日期	修订记录
2022-12-09	第一次正式发布。