

商品组件

onappear算法

懒加载思路



by eric wen

分析UI

分析UI

I 商品组件有三种样式，大、小、横

I 点击进入商品详情

I 特殊需求：懒加载，在商品真正要被暴露在视口时，才触发组件的UI渲染

核心技术：onappear dom在视口前暴露事件

特殊需求讲解：

I 节省用户流星，用户大多数情况下打开页面并不下拉到后面，则不渲染图片等

I 性能，前端越多div结构，越影响流畅性，没暴露在视口的dom可以最大程度保持结构简单，直到将来暴露再渲染

I 下拉加载，这是h5应用的最常见交互

I 数据收集，现代应用都要收集用户数据，用于大数据分析，当元素曝光时，主动上报用户相关数据

组件设计

Props	Type 控制样式 1竖大图 2横 3 竖小图		

自定义Hooks：useAppear

监控当前组件是否暴露在视口。

Props	Threshold 控制组件出现在视口多少时触发appear	rootMargin控制组件离视口还有多少距离时触发，用此参数可以提前触发	Once 控制是否重复触发，根据使用场景，
Return	Ref 和引用组件的dom元素相连，在内部传递给监控api	IsAppear 触发则true	

组件开发

用type控制样式

useAppear hook

使用新的api:IntersectionObserver

此API的兼容性可以。



相比早期用getBoundingClientRect，要更加精准，使用方便，性能更好。

因为getBoundingClientRect要监听滚动事件，滚动事件的触发最小周期受js引擎和渲染时机的影响。

懒加载

懒加载主要控制两种资源。

I 控制dom体量，在isAppear成true时，才展开组件dom结构。

I 控制http请求，减轻服务端压力，在isAppear成true时，才调用数据