

模块事件

一、上架商品

说明： 当商品上架时，会触发该事件

Event 事件： `ProductEvents.OnSaleProduct`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>productData</code>	商品数据	<code>ProductData</code>

示例：

```
productTarget.eventEmitter.on(ProductEvents.OnSaleProduct, (data) => {
  console.log('商品数据', data.productData);
});
```

二、下架商品

说明： 当商品下架时，会触发该事件

Event 事件： `ProductEvents.OffSaleProduct`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>productData</code>	商品数据	<code>ProductData</code>

示例：

```
productTarget.eventEmitter.on(ProductEvents.OffSaleProduct, (data) => {
  console.log('商品数据', data.productData);
});
```

三、添加商品

说明：当商品添加时，会触发该事件

Event 事件： `ProductEvents.AddProduct`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>productData</code>	商品数据	<code>ProductData</code>

示例：

```
productTarget.eventEmitter.on(ProductEvents.AddProduct, (data) => {
  console.log('商品数据', data.productData);
});
```

四、删除商品

说明：当商品删除时，会触发该事件

Event 事件： `ProductEvents.DeleteProduct`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>productId</code>	商品 id	<code>number</code>

示例：

```
productTarget.eventEmitter.on(ProductEvents.DeleteProduct, (data) => {
  console.log('商品 id', data.productId);
});
```

五、修改商品

说明：当商品修改时，会触发该事件

Event 事件： `ProductEvents.ChangeProduct`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>productData</code>	商品数据	<code>ProductData</code>

示例：

```
productTarget.eventEmitter.on(ProductEvents.ChangeProduct, (data) => {
  console.log('商品数据', data.productData);
});
```

六、上移或下移商品

说明： 当商品上移或下移时，会触发该事件

Event 事件： `ProductEvents.MoveProduct`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>targetProducts</code>	发生改变的两个商品	<code>[ProductData, ProductData]</code>

示例：

```
productTarget.eventEmitter.on(ProductEvents.MoveProduct, (data) => {
  console.log('发生改变的两个商品', data.targetProducts);
});
```

七、修改商品排序

说明： 当商品排序修改时，会触发该事件

Event 事件： `ProductEvents.ChangeProductRank`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
productData	商品数据	ProductData

示例：

```
productTarget.eventEmitter.on(ProductEvents.ChangeProductRank, (data) => {
  console.log('商品数据', data.productData);
});
```

八、商品售罄

说明：当商品售罄时，会触发该事件

Event 事件：ProductEvents.ProductSoldOut

回调参数：Object 对象，详细类型说明如下

属性名	说明	类型
productId	商品 id	number

示例：

```
productTarget.eventEmitter.on(ProductEvents.ProductSoldOut, (data) => {
  console.log('商品 id', data.productId);
});
```

九、推送商品

说明：当商品推送时，会触发该事件

Event 事件：ProductEvents.PushProduct

回调参数：Object 对象，详细类型说明如下

属性名	说明	类型
productData	商品数据	ProductData

属性名	说明	类型
pushRule	推送方式	ProductPushRule

示例：

```
productTarget.eventEmitter.on(ProductEvents.PushProduct, (data) => {
  console.log('商品数据', data.productData);
});
```

十、取消推送商品

说明： 当商品取消推送时，会触发该事件

Event 事件： ProductEvents.CancelPushProduct

回调参数： Object 对象，详细类型说明如下

属性名	说明	类型
productId	商品 id	number

示例：

```
productTarget.eventEmitter.on(ProductEvents.CancelPushProduct, (data) => {
  console.log('商品 id', data.productId);
});
```

十一、商品库列表开关

说明： 当商品库列表开关时，会触发该事件

Event 事件： ProductEvents.ProductListEnabled

回调参数： Object 对象，详细类型说明如下

属性名	说明	类型
enabled	开启或关闭	boolean

示例：

```
productTarget.eventEmitter.on(ProductEvents.ProductListEnabled, (data) => {
  console.log('开启或关闭', data.enabled);
});
```

十二、刷新当前商品列表

说明： 当商品列表刷新时， 会触发该事件

Event 事件： `ProductEvents.RefreshProductList`

示例：

```
productTarget.eventEmitter.on(ProductEvents.RefreshProductList, (data) => {
  console.log('商品列表', data.productList);
});
```

十三、商品点击事件

说明： 当商品点击时， 会触发该事件

Event 事件： `ProductEvents.ProductClick`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>productId</code>	商品 id	<code>number</code>
<code>name</code>	商品名称	<code>string</code>
<code>productType</code>	商品类型	<code>ProductType</code>

示例：

```
productTarget.eventEmitter.on(ProductEvents.ProductClick, (data) => {
  console.log('商品 id', data.productId);
});
```

十四、商品点击次数统计事件

说明： 当点击次数发生改变时，会触发该事件

Event 事件： `ProductEvents.ProductClickTimes`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>productId</code>	商品 id	<code>number</code>
<code>name</code>	商品名称	<code>string</code>
<code>productType</code>	商品类型	<code>ProductType</code>
<code>times</code>	点击次数	<code>number</code>

示例：

```
productTarget.eventEmitter.on(ProductEvents.ProductClickTimes, (data) => {
  console.log('商品 id', data.productId);
});
```