

使用方式

一、获取商品标签列表

Api 方法： `getProductTagList(params?: PageOptions): Promise<PageContent<ProductTag>>`

参数说明：

- params：获取参数， `PageOptions` 类型，选传，默认 `{}`

返回值说明： `Promise<PageContent<ProductTag>>` 类型

示例：

```
const data = await productTarget.getProductTagList({ pageNumber: 1 });
console.log('标签列表', data.contents);
```

二、获取商品列表

该方法可通过 **商品序号**、**标签 id + 关键词** 两种方式获取商品列表，

Api 方法： `getProductList(params?: GetProductListParams): Promise<ProductListResult>`

参数说明：

- params：获取参数， `GetProductListParams` 类型，选传，默认 `{}`，详细类型说明如下

参数名	说明	类型	必须	默认值
<code>count</code>	获取数量	<code>number</code>	否	<code>10</code>
<code>displayNumber</code>	商品序号	<code>number</code>	否	-
<code>tagId</code>	标签 id	<code>number</code>	否	-
<code>keyword</code>	关键词	<code>string</code>	否	-
<code>page</code>	页数，仅传入	<code>number</code>	否	-

返回值说明： `Promise<ProductListResult>` 类型，详细类型说明如下

属性名	说明	类型
contents	商品列表	ProductData[]
total	总数	number
toppingProduct	置顶商品	ProductData

示例：

```
// 通过商品序号获取：获取 12 号商品之后的商品列表
const data = await productTarget.getProductList({ displayNumber: 12 });

// 通过标签 id 或关键词获取：获取标签 id 为 8 的商品列表，并根据关键词 '水果' 进行搜索，页数为第 2 页
const data = await productTarget.getProductList({ tagId: 8, keyword: '水果', page: 2 });
```

三、获取商品列表（发起端）

Api 方法：`getLaunchProductList(params: GetLaunchProductListParams): Promise<LaunchProductListResult>`

参数说明：

- params：获取参数，`GetLaunchProductListParams` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
pageNumber	-	number	否	-
pageSize	-	number	否	-
keyword	-	string	否	-
status	在架状态	ProductStatus	否	-

返回值说明：`Promise<LaunchProductListResult>` 类型，详细类型说明如下

属性名	说明	类型
pageNumber	当前页数	number
pageSize	每页数量	number

属性名	说明	类型
totalItems	总记录数	number
totalPages	总页数	number
contents	内容列表	ProductData[]
upCount	上架数量	number
downCount	下架数量	number

四、获取商品数据

Api 方法： `getProductData(productId: number): Promise<ProductData>`

参数说明：

- productId: 商品 id, `number` 类型, 必传

返回值说明： `Promise<ProductData>` 类型

五、发送商品点击事件

当观众点击商品时, 调用该方法发送商品点击事件

Api 方法： `sendProductClickEvent(params: SendProductClickParams): Promise<void>`

参数说明：

- params: 发送参数, `SendProductClickParams` 类型, 必传, 详细类型说明如下

参数名	说明	类型	必须	默认值
productId	商品 id	number	是	-
name	商品名称	string	是	-
productType	商品类型	ProductType	是	-

示例：

```
await productTarget.sendProductClickEvent({
  productId: 123,
  name: '商品名称',
  productType: 'normal',
});
```

六、关闭当前正在推送中的商品

当需要关闭 UI 层的推送节点时，调用该方法关闭当前正在推送中的商品

Api 方法： `closeProductPush(): void`

示例：

```
await productTarget.closeProductPush();
```

七、获取商品讲解开关

获取当前频道是否开启商品讲解

Api 方法： `getProductExplainEnabled(): Promise<boolean>`

返回值说明： `Promise<boolean>` 类型

示例：

```
const enabled = await productTarget.getProductExplainEnabled();
console.log('商品讲解开关', enabled);
```

八、生成商品讲解页面地址

Api 方法： `generateProductExplainPageUrl(params?: T): Promise<null | string>`

参数说明：

- params: url 携带参数， `T` 类型，选传

返回值说明： `Promise<null | string>` 类型

示例：

```
const url = await productTarget.generateProductExplainPageUrl({ productId: 123 });
console.log('商品讲解页面地址', url);
```

九、获取当前频道是否开启点击"外链购买"商品的封面/标题区域直接跳转外链

Api 方法： `getOutLinkProductRedirectEnabled(): Promise<boolean>`

返回值说明： `Promise<boolean>` 类型

示例：

```
const enabled = await productTarget.getOutLinkProductRedirectEnabled();
console.log('商品点击直接跳转外链开关', enabled);
```

十、获取商品支付订单是否开启

Api 方法： `getProductPayOrderEnabled(): Promise<boolean>`

返回值说明： `Promise<boolean>` 类型

示例：

```
const enabled = await productTarget.getProductPayOrderEnabled();
console.log('商品支付订单是否开启', enabled);
```

十一、获取商品热卖配置

Api 方法： `getProductHotSaleConfig(): Promise<undefined | ProductHotSaleConfig>`

返回值说明： `Promise<undefined | ProductHotSaleConfig>` 类型

示例：

```
const config = await productTarget.getProductHotSaleConfig();
console.log('商品热卖配置', config);
```

十二、获取当前推送中的商品数据

Api 方法: `getCurrentPushingProduct(productId?: number): Promise<undefined | ProductData>`

参数说明:

- productId: 可选, 指定商品ID, 如果不提供则获取当前配置中的推送商品, `number` 类型, 选传

返回值说明: `Promise<undefined | ProductData>` 类型

示例:

```
const productData = await productTarget.getCurrentPushingProduct();
console.log('当前推送中的商品数据', productData);
```

十三、开始商品讲解

Api 方法: `startExplain(productId: number): undefined | Promise<void>`

参数说明:

- productId: 商品 id, `number` 类型, 必传

返回值说明: `undefined | Promise<void>` 类型

示例:

```
await productTarget.startExplain(123);
```

十四、重新开始商品讲解

Api 方法: `restartExplain(productId: number): undefined | Promise<void>`

参数说明:

- productId: 商品 id, `number` 类型, 必传

返回值说明: `undefined | Promise<void>` 类型

示例:

```
await productTarget.restartExplain(123);
```

十五、结束商品讲解

Api 方法: `endExplain(productId: number): undefined | Promise<void>`

参数说明:

- productId: 商品 id, `number` 类型, 必传

返回值说明: `undefined | Promise<void>` 类型

示例:

```
await productTarget.endExplain(123);
```

十六、删除商品讲解

Api 方法: `deleteExplain(productId: number): Promise<void>`

参数说明:

- productId: 商品 id, `number` 类型, 必传

示例:

```
await productTarget.deleteExplain(123);
```

十七、获取当前正在讲解的商品 id

Api 方法: `getCurrentProductExplain(): Promise<null | number>`

返回值说明: `Promise<null | number>` 类型

示例:

```
const productId = await productTarget.getCurrentProductExplain();
console.log('当前正在讲解的商品 id', productId);
```

十八、商品开价

Api 方法：`openProductPrice(productId: number): Promise<void>`

参数说明：

- productId：商品 id，`number` 类型，必传

示例：

```
await productTarget.openProductPrice(123);
```

十九、商品隐藏价格

Api 方法：`hideProductPrice(productId: number): Promise<void>`

参数说明：

- productId：商品 id，`number` 类型，必传

示例：

```
await productTarget.hideProductPrice(123);
```

二十、获取商品设置

Api 方法：`getProductSetting(): Promise<ProductSetting>`

返回值说明：商品设置，`Promise<ProductSetting>` 类型，详细类型说明如下

属性名	说明	类型
<code>productExplainEnabled</code>	商品讲解开关	<code>boolean</code>
<code>productExplainingAutoPushAndSticky</code>	商品讲解自动推送置顶	<code>boolean</code>
<code>productPushRule</code>	商品推送方式	<code>ProductPushRule</code> <code>'chooseCard'</code>
<code>pushingProduct</code>	推送中的商品	<code>Object</code>

二十一、推送商品

Api 方法: `pushProduct(productId: number, rule?: ProductPushRule): Promise<void>`

参数说明:

- productId: 商品 id, `number` 类型, 必传
- rule: 推送规则, `ProductPushRule` 类型, 选传

示例:

```
await productTarget.pushProduct(123, ProductPushRule.SmallCard);
```

二十二、取消推送商品

Api 方法: `cancelPushProduct(productId: number): Promise<void>`

参数说明:

- productId: 商品 id, `number` 类型, 必传

示例:

```
await productTarget.cancelPushProduct(123);
```

二十三、批量删除商品

Api 方法: `batchDeleteProduct(productIds: number[]): Promise<void>`

参数说明:

- productIds: 商品 id 列表, `number[]` 类型, 必传

示例:

```
await productTarget.batchDeleteProduct([123, 456]);
```

二十四、批量上架商品

Api 方法： `batchOnShelfProduct(productIds: number[]): Promise<void>`

参数说明：

- productIds: 商品 id 列表, `number[]` 类型, 必传

示例：

```
await productTarget.batchOnShelfProduct([123, 456]);
```

二十五、批量下架商品

Api 方法： `batchOffShelfProduct(productIds: number[]): Promise<void>`

参数说明：

- productIds: 商品 id 列表, `number[]` 类型, 必传

示例：

```
await productTarget.batchOffShelfProduct([123, 456]);
```

二十六、判断当前用户是否支持直接购买商品

Api 方法： `getSupportToDirectBuy(): Promise<boolean>`

返回值说明： `Promise<boolean>` 类型

示例：

```
const supported = await productTarget.getSupportToDirectBuy();  
console.log('是否支持直接购买', supported);
```