1. **Category Explore操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”category”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {category: 子分类数组}

子分类数组 = [ 子分类对象, 子分类对象, … ]

子分类对象 = { id : 整数, name : 字串, active: 布尔值, image: 字串, typeId: 整数, changeType: 布尔值, hasProduct: 布尔值, hasSubCategory: 布尔值 }

1. **Product Explore操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”product”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {product: 商品数组}

商品数组= [商品对象, 商品对象, … ]

商品对象 = { id : 整数, name : 字串}

1. **Category Remove操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”removecategory”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {category: 子分类数组}

子分类数组 = [ 子分类对象, 子分类对象, … ]

子分类对象 = { id : 整数, name : 字串, active: 布尔值, image: 字串, typeId: 整数, changeType: 布尔值 }

1. **Product Remove操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”removeproduct”，json=消息

消息 = [ {id: 整数}, [请求对象, … ] ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {product: 商品数组}

商品数组= [商品对象, 商品对象, … ]

商品对象 = { id : 整数, name : 字串}

1. **Category Query Image操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”categoryimage”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {image: image数组}

image数组 = [字串(image路径), 字串(image路径), … ]

1. **Product Query Image操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”productimage”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {image: image数组}

image数组 = [image对象, image对象, … ]

image对象 = {name: 字串(显示名), image: 字串(真实路径)}

字串(image路径), 字串(image路径), …

1. **Category Query Names操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”categoryname”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {name: Name数组}

Name数组 = [Name对象, Name对象, … ]

Name对象 = { langId : 整数, header : 字串, code: 字串, name : 字串}

1. **Product Query Names操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”productname”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {name: Name数组}

Name数组 = [Name对象, Name对象, … ]

Name对象 = { langId : 整数, header : 字串, code: 字串, name : 字串}

1. **Query操作（作废）：**
   1. URL: admin/category/query
   2. Client->server

消息 = [ 请求对象 ]

请求对象 = { id : 整数, parentId : 整数 }

* 1. Server -> client

消息 = [Category对象]

Category对象 = { id : 整数, parentId : 整数, addTime字串: , updateTime : 字串, active : 布尔, image : 字串, displayOrder : 整数, general : [general对象, general对象, … ] }

general对象 = { langId : 整数, header : 字串, name : 字串, description : 字串 }

1. **Category Save操作：**
   1. URL: admin/product/savecategory
   2. Client->server

请求方法: POST

请求参数: action=” savecategory”(url如果没有指定的情况下必须通过此参数指明操作)，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数, parentId : 整数, active : 布尔, typeId: 整数, image : [字串, 字串, 字串, … ], name : [name对象, name对象, … ] , displayOrder: [displayOrder对象, displayOrder对象, displayOrder对象, … ]}

name对象 = { langId : 整数, name : 字串}

displayOrder对象, = {id : 整数, displayOrder : 整数}

* 1. Server->client

消息 = {category: 子分类数组}

子分类数组 = [ 子分类对象, 子分类对象, … ]

子分类对象 = { id : 整数, name : 字串, active: 布尔值}

1. **Product Save操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=” saveproduct”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数, categoryId : 整数, active : 布尔, image : [字串, 字串, 字串, … ], name : [name对象, name对象, … ], parameter: 字串, extensionProperty: {ep1: 整数, ep2: 整数, … , ep5: 整数} }

name对象 = { langId : 整数, name : 字串}

* 1. Server->client

消息 = {product: 商品数组}

商品数组= [商品对象, 商品对象, … ]

商品对象 = { id : 整数, name : 字串}

1. **Product SaveDisplayOrder操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=” saveproductdisplayorder”，json=消息

消息 = [ 请求对象 ]

请求对象 = {categoryId : 整数, displayOrder: [displayOrder对象, displayOrder对象, displayOrder对象, … ]}

displayOrder对象, = {id : 整数, displayOrder : 整数}

* 1. Server->client

消息 = {product: 商品数组}

商品数组= [商品对象, 商品对象, … ]

商品对象 = { id : 整数, name : 字串}

1. **Product Query Type操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”producttype”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {typeId: 整数}

1. **Product Query Parameter操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”productparameter”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 =[code(字串), [参数组对象, 参数组对象, … ] ]

1. **Product Save Parameter操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”saveproductparameter”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数, categoryId : 整数, parameter: 字串}

* 1. Server->client

消息 =[code(字串), [参数组对象, 参数组对象, … ] ]

1. **Product Query ExtensionProperty操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”productextensionproperty”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = {extensionProperty: [整数, 整数, 整数, 整数, 整数(分别为EPn的值, n = 1,2,3,4,5)]}

EPn值：-1表示未选中，其余非负整数表示选中第几项

1. **Product Save ExtensionProperty操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”saveproductextensionproperty”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数, categoryId : 整数, extensionProperty: [整数, 整数, 整数, 整数, 整数(分别为EPn的值, n = 1,2,3,4,5)]}

* 1. Server->client

消息 = {extensionProperty: [整数, 整数, 整数, 整数, 整数(分别为EPn的值, n = 1,2,3,4,5)]}

EPn值：-1表示未选中，其余非负整数表示选中第几项

1. **Product Query Option操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”productoption”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

消息 = { isOptionUsing: 布尔值, option: [option对象, option对象, … ], goods: [goods对象, goods对象, goods对象, …] }

option对象 = { group: {id: 整数, name: 字串}, value: [{id: 整数, name: 字串}, {id: 整数, name: 字串}, {id: 整数, name: 字串}, … ] }

goods对象 = {id: 整数, store : 整数, bn: 字串, storePlace: 字串, weight: 双精度, cost: 双精度, marketPrice: 双精度, price: 双精度, option: 待定 }

1. **Product Open Option操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=” openproductoption”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

同18

1. **Product Close Option操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=” closeproductoption”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

同18

1. **Product Save Option操作（作废）：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=”saveproductoption”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数, categoryId : 整数, isOptionUsing: 布尔值, option: [optionSetting对象, optionSetting对象, … ] }

optionSetting对象 = { groupId: 整数, value: [整数, 整数, 整数, … ] }

* 1. Server->client

消息 = { isOptionUsing: 布尔值, option: [optionSetting对象, optionSetting对象, … ]}

1. **Product Add Option Group操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=” addproductoptiongroup”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数, groupId : 整数 }

* 1. Server->client

消息 = { groupId: 整数(-1表示操作失败), name: 字串, error: 字串}

1. **Product Remove Option Group操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=” removeproductoptiongroup”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数, groupId : 整数 }

* 1. Server->client

消息 = { groupId: 整数(-1表示操作失败), name: 字串, error: 字串}

1. **Product Add Option Value操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=” closeproductoption”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

同18

1. **Product Remove Option Value操作：**
   1. URL: admin/product/
   2. Client->server

请求方法: POST

请求参数: action=” closeproductoption”，json=消息

消息 = [ 请求对象 ]

请求对象 = { id : 整数 }

* 1. Server->client

同18