前言

以下API的根目录路径均为/xxx/store, xxx为context-path

假设context-path是is-dev,则例如第一个api的地址为: <http://localhost:8080/is-dev/store/campaign/1/categories>

所有API均返回固定的结构,举例如下:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:**…

}

其中status==0时,为无错误情况.如果非0,则需要进行特定处理.

status==-100时,表示用户未登录,需要跳转去登陆相关页面

status==-1时候,需要显示用户无权限的页面

status==-2的时候,需要显示活动倒计时提示(相关API之后整理)

status==-3的时候,需要显示活动已结束提示(相关API之后整理)

其他错误码目前还未完全确定,

data的内容则为实际数据,之后API中提到的返回data也是指的这个字段的内容

~~商店的逻辑层级从大到小是~~

~~campaign - type – cat（category） - good – product~~ 待重构

所有返回对象中的orderNum是排序顺序。

~~所有返回对象中的status为状态,只显示给用户status==1的,status==0的为禁用的~~

good是展示给用户看的每个宝贝

product是具体每个宝贝里，通过不同规格组合确定的每个独立的商品

price和originPrice,如果originPrice不为空,则需要显示原价,否则不显示原价.

调试方法

目前,staging服务器的API可以直接调用,地址为

<http://is.yimlinkapp.com/is-dev/>store/...

由于方便起见,我们之后会修改项目地址,跟生产服务器统一为

<http://is.yimlinkapp.com/is/>store/...

在调用API之前,需要先调用:

<http://is.yimlinkapp.com/is-dev/auth/loginRequestInternalDebug?account=yimlink.defia&password=1>

进行模拟登陆,才能得到正确的返回结果.

如:

http://is.yimlinkapp.com/is-dev/store/campaign/45/info

正文

1. 获取活动列表

@RequestMapping(value = "/campaigns", method = RequestMethod.GET)

返回内容:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **[**

**{**

"id"**:** 1**,**

"status"**:** 1**,**

"name"**:** "Celine Family Sales Test"**,**

"pic"**:**"http://www.baidu.com/1.jpg"**,**

"brandId"**:** 1**,**

"brandName"**:** "Celine"**,**

"begin"**:** 1517064934000**,**

"end"**:** 1517064938000**,**

"ruleId"**:** 1**,**

"campaignType"**:** 1

**}**

**]**

**}**

1. 获取活动信息

@RequestMapping(value = "/campaign/{id}/info", method = RequestMethod.GET)

返回内容:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **{**

"id"**:** 1**,**

"status"**:** 1**,**

"name"**:** "Celine Family Sales Test"**,**

"banner"**:**"http://www.baidu.com/1.jpg"**,**

"brandId"**:** 1**,**

"brandName"**:** "Celine"**,**

"begin"**:** 1517064934000**,**

"end"**:** 1517064938000**,**

"ruleId"**:** 1**,**

"campaignType"**:** 1

**}**

**}**

1. 获取当前用户当前活动额度等信息

@RequestMapping(value = "/campaign/{id}/userinfo", method = RequestMethod.GET)

返回内容:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **{**

"name"**:** "Grace Henderson"**,**

"head"**:** "http://www.baidu.com/head.jpg"**,**

"point"**:** 12000**,**

"brand"**:**"GIVENCHY"**,**

"joinDate"**:**"2019/03/14"

**}**

**}**

4.获取指定campaign下的所有分类及类别(待重构)

@RequestMapping(value = "/campaign/{id}/categories", method = RequestMethod.GET)

{id}为campaign的id

返回内容:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **{**

"types"**:** **[**

**{**

"id"**:** 4**,**

"status"**:** 1**,**

"orderNum"**:** 0**,**

"name"**:** "LG"**,**

"enName"**:** **null,**

"campaignId"**:** 1

**}**

**],**

"cats"**:** **[**

**{**

"id"**:** 3**,**

"status"**:** 1**,**

"name"**:** "CatA"**,**

"enName"**:** **null,**

"campaignId"**:** 1**,**

"productTypeId"**:** 4**,**

"orderNum"**:** 0

**}**

**]**

**}**

**}**

.type为1级分类,cat为二级分类,cat的productTypeId为其根type的id.

目前系统设计上只有2级分类.需要前端进行拼接成2级树状结构

5.获取type下的所有宝贝

@RequestMapping(value = "/type/{id}/goods", method = RequestMethod.GET)

{id}为typeId

返回内容:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **[**

**{**

"id"**:** 5**,**

"status"**:** 0**,**

"orderNum"**:** 0**,**

"name"**:** "LG\_CatA\_NameA"**,**

"enName"**:** "LG\_CatA\_NameA\_en"**,**

"price"**:** 2000.00**,**

"originPrice"**:** 3000.00**,**

"pic"**:** **null,**

"campaignId"**:** 1**,**

"productTypeId"**:** 4**,**

"productCategoryId"**:** 3

**},**

**{**

"id"**:** 6**,**

"status"**:** 1**,**

"orderNum"**:** 0**,**

"name"**:** "LG\_CatA\_NameA"**,**

"enName"**:** "LG\_CatA\_NameA\_en"**,**

"price"**:** 2000.00**,**

"originPrice"**:** 3000.00**,**

"pic"**:** **null,**

"campaignId"**:** 1**,**

"productTypeId"**:** 4**,**

"productCategoryId"**:** 3

**}**

**]**

**}**

包含宝贝中英文名，图片地址, 价格,原价等信息

6. 获取cat下所有宝贝

@RequestMapping(value = "/category/{id}/goods", method = RequestMethod.GET)

返回同上

7.获取campaign下所有宝贝

@RequestMapping(value = "/campaign/{id}/goods", method = RequestMethod.GET)

返回同上

8.单个good信息查询

@RequestMapping(value = "/good/{id}", method = RequestMethod.GET)

Data内容同之前查看good的几个api的单元素

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **{**

"id"**:** 12**,**

"status"**:** 1**,**

"orderNum"**:** 0**,**

"name"**:** "LG\_CatA\_NameA"**,**

"enName"**:** "LG\_CatA\_NameA\_en"**,**

"price"**:** 2000.00**,**

"originPrice"**:** **null,**

"pic"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/f33074de-a50d-4ff7-aa46-8ac110a1747f.jpg?x-oss-process=image/resize,w\_320"**,**

"campaignId"**:** 1**,**

"campaignType"**:** **null,**

"productTypeId"**:** 5**,**

"productCategoryId"**:** 4

**}**

**}**

9.获取某个宝贝下的所有不同商品

@RequestMapping(value = "/good/{id}/products", method = RequestMethod.GET)

{id}为good id

返回内容：

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **[**

**{**

"id"**:** 1**,**

"status"**:** 1**,**

"orderNum"**:** 0**,**

"campaignId"**:** 1**,**

"goodId"**:** 6**,**

"sku"**:** "223456"**,**

"name"**:** "LG\_CatA\_NameA"**,**

"enName"**:** "LG\_CatA\_NameA\_en"**,**

"productTypeId"**:** 4**,**

"productCategoryId"**:** 3**,**

"price"**:** 2000.00**,**

"originPrice"**:** **null,**

"picUrl"**:** **null,**

"spec1"**:** "Black"**,**

"spec2"**:** "M"**,**

"description"**:** "LG\_CatA\_NameA\_223456\_desc"**,**

"enDescription"**:** "LG\_CatA\_NameA\_223456\_desc\_en"

**},**

**{**

"id"**:** 2**,**

"status"**:** 1**,**

"orderNum"**:** 10**,**

"campaignId"**:** 1**,**

"goodId"**:** 6**,**

"sku"**:** "123456"**,**

"name"**:** "LG\_CatA\_NameA"**,**

"enName"**:** "LG\_CatA\_NameA\_en"**,**

"productTypeId"**:** 4**,**

"productCategoryId"**:** 3**,**

"price"**:** 1000.00**,**

"originPrice"**:** **null,**

"picUrl"**:** **null,**

"spec1"**:** "White"**,**

"spec2"**:** "M"**,**

"description"**:** "LG\_CatA\_NameA\_123456\_desc"**,**

"enDescription"**:** "LG\_CatA\_NameA\_123456\_desc\_en"

**}**

**]**

**}**

其中spec1和spec2为多规格的字段，1个宝贝会有0-2个规格，每个规格会有若干种不同选择,如果某规格不存在则对应规格字段为null，前端也应该不显示该规格的选择.

后端返回的不同商品的列表为平铺的,需要前端进行整理并组合出每组规格组合对应的库存,图片等信息..

10.获取某个宝贝下所有不同商品的库存情况.

@RequestMapping(value = "/good/{id}/stock", method = RequestMethod.GET)

{id}为good id

返回内容:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **[**

**{**

"id"**:** 1**,**

"campaignId"**:** 1**,**

"goodId"**:** 7**,**

"productId"**:** 3**,**

"productSku"**:** "223456"**,**

"productName"**:** "LG\_CatA\_NameA"**,**

"total"**:** 15**,**

"now"**:** 15

**},**

**{**

"id"**:** 2**,**

"campaignId"**:** 1**,**

"goodId"**:** 7**,**

"productId"**:** 4**,**

"productSku"**:** "123456"**,**

"productName"**:** "LG\_CatA\_NameA"**,**

"total"**:** 15**,**

"now"**:** 15

**}**

**]**

**}**

Now为显示给用户看的剩余库存.库存为0则无法购买

11.获取某个宝贝下所有不同商品的图片.

@RequestMapping(value = "/good/{id}/pictures", method = RequestMethod.GET)

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **[**

**{**

"id"**:** 13**,**

"campaignId"**:** 1**,**

"goodId"**:** 12**,**

"productId"**:** 3**,**

"orderNum"**:** 2**,**

"url160"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/f33074de-a50d-4ff7-aa46-8ac110a1747f.jpg?x-oss-process=image/resize,w\_160"**,**

"url320"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/f33074de-a50d-4ff7-aa46-8ac110a1747f.jpg?x-oss-process=image/resize,w\_320"**,**

"url640"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/f33074de-a50d-4ff7-aa46-8ac110a1747f.jpg?x-oss-process=image/resize,w\_640"**,**

"url1280"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/f33074de-a50d-4ff7-aa46-8ac110a1747f.jpg?x-oss-process=image/resize,w\_1280"**,**

"url2560"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/f33074de-a50d-4ff7-aa46-8ac110a1747f.jpg?x-oss-process=image/resize,w\_2560"

**},**

**{**

"id"**:** 14**,**

"campaignId"**:** 1**,**

"goodId"**:** 12**,**

"productId"**:** 3**,**

"orderNum"**:** 1**,**

"url160"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/59f787c3-86b9-49fd-b3af-52b5d1735f45.jpg?x-oss-process=image/resize,w\_160"**,**

"url320"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/59f787c3-86b9-49fd-b3af-52b5d1735f45.jpg?x-oss-process=image/resize,w\_320"**,**

"url640"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/59f787c3-86b9-49fd-b3af-52b5d1735f45.jpg?x-oss-process=image/resize,w\_640"**,**

"url1280"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/59f787c3-86b9-49fd-b3af-52b5d1735f45.jpg?x-oss-process=image/resize,w\_1280"**,**

"url2560"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/59f787c3-86b9-49fd-b3af-52b5d1735f45.jpg?x-oss-process=image/resize,w\_2560"

**},**

**{**

"id"**:** 15**,**

"campaignId"**:** 1**,**

"goodId"**:** 12**,**

"productId"**:** 4**,**

"orderNum"**:** 1**,**

"url160"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/d569d23e-29fc-4f5c-8c69-7489a9ef058e.jpg?x-oss-process=image/resize,w\_160"**,**

"url320"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/d569d23e-29fc-4f5c-8c69-7489a9ef058e.jpg?x-oss-process=image/resize,w\_320"**,**

"url640"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/d569d23e-29fc-4f5c-8c69-7489a9ef058e.jpg?x-oss-process=image/resize,w\_640"**,**

"url1280"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/d569d23e-29fc-4f5c-8c69-7489a9ef058e.jpg?x-oss-process=image/resize,w\_1280"**,**

"url2560"**:** "http://yimtest.oss-cn-shanghai.aliyuncs.com/d569d23e-29fc-4f5c-8c69-7489a9ef058e.jpg?x-oss-process=image/resize,w\_2560"

**}**

**]**

**}**

由于每个商品可能有多个图片,之前提供的API中返回的图片地址,仅用于单个图片的场合显示用

在商品详情页选择了具体的规格,并根据规格组合对应的product Id找到对应的图片后,需进行轮播

12.获取购物车数量.

@RequestMapping(value = "/cartNum", method = RequestMethod.GET)

返回:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **{**

“qty”:5

}

}

13.获取购物车详情

@RequestMapping(value = "/cart", method = RequestMethod.GET)

返回:

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **[{**

"id"**:**1**,**

"productId"**:**4**,**

"qty"**:**2**,**

"name"**:**"LG\_CatA\_NameA"**,**

"enName"**:**"LG\_CatA\_NameA\_en"**,**

"pic"**:**"http://www.baidu.com/1.jpg"**,**

"price"**:**2000.00**,**

"originPrice"**:** **null,**

"spec1"**:**"White"**,**

"spce2"**:**"M"

**}]**

**}**

14.设置购物车指定productId的数量

@RequestMapping(value = "/cart/modify", method = RequestMethod.POST)

postBody包含如下两个param:

productId

qty

返回内容:

**{**

"status"**:** 0**,**

"message"**:** "success"

**}**

15.获取用户各订单状态统计

@RequestMapping(value = "/order/stats", method = RequestMethod.GET)

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **{**

"all"**:** 4**,**

"paid"**:** 2**,**

"transport"**:** 2**,**

"unpaid"**:** 0

**}**

**}**

16.获取用户订单列表

@RequestMapping(value = "/order/list", method = RequestMethod.GET)

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **[{**

"id"**:**1**,**

"campaignName"**:**"Trail Goods 2018"**,**

"status"**:**5**,**

"orderDate"**:**"2017-12-24 9:05:23"**,**

"amount"**:**2000.00**,**

"items"**:[{**

"name"**:**"LG\_CatA\_NameA"**,**

"enName"**:**"LG\_CatA\_NameA\_en"**,**

"price"**:**2000.00**,**

"qty"**:**1**,**

"spec1"**:**"White"**,**

"spec2"**:**"M",

"pic"**:**"http://www.baidu.com/1.jpg"

**}]**

**}]**

**}**

17.获取订单详情

@RequestMapping(value = "/order/{id}/info", method = RequestMethod.GET)

**{**

"status"**:** 0**,**

"message"**:** "success"**,**

"data"**:** **{**

"id"**:**1**,**

"campaignName"**:**"Trail Goods 2018"**,**

"status"**:**5**,**

"orderDate"**:**"2017-12-24 09:05:23"**,**

"orderNo"**:**"20171214090523-123456"**,**

"amount"**:**2000.00**,**

"payMethod"**:**"微信支付"**,**

"mobile"**:**"13012345678"**,**

"payDate"**:**"2017-12-24 09:08:08"**,**

"userName"**:**"Gracee Henderson"

"items":**[{**

"name"**:**"LG\_CatA\_NameA"**,**

"enName"**:**"LG\_CatA\_NameA\_en"**,**

"price"**:**2000.00**,**

"qty"**:**1**,**

"spec1"**:**"White"**,**

"spec2"**:**"M"**,**

"pic"**:**"http://www.baidu.com/1.jpg"

**}]**

**}**

**}**