

console-center-aggregate 接口文档

商城套餐版本下单

接口地址

POST /v3/console/mall-package-version/create-order

接口说明

根据商城套餐版本 ID 创建订单。

当前接口由 console-center-aggregate 提供，内部先调用 MallApiClient#getVersionDetail 获取版本详情，再按商城版本信息转换为续费下单模型 SubmitRenewalOrderReq，最终复用支付侧 /api/payment/submit_renewal_order 创建订单。

当前实现按新购处理，订单类型固定为 0。

请求参数

参数名	类型	必填	说明
id	Integer	是	商城套餐版本 ID

请求示例

```
{
  "id": 1001
}
```

版本详情转换规则

来源字段	目标字段	说明
id	productId	订单产品 ID，转字符串

来源字段	目标字段	说明
<code>name</code>	<code>productName / packageName</code>	订单名称、套餐名称
<code>crm_product_id</code>	<code>productInfo.productCode</code>	CRM 商品编码
<code>description</code>	<code>productInfo.packageDesc</code>	套餐描述
<code>unit_price</code>	<code>productAmount / productInfo.amount</code>	金额按元解析, <code>productAmount</code> 转成分
<code>price_cycle + price_cycle_value</code>	<code>startDate / endDate / validity / validityUnit</code>	开始时间取当前日期, 结束时间按周期计算
<code>resource_quotas.type = main_package</code>	<code>billingPlanCode / quantity / billingUnit</code>	主套餐资源信息
<code>resource_quotas.type = function</code>	<code>functionList</code>	转为 clearing 可识别的功能列表

说明：

- `startDate` 取当前日期, 格式 `yyyy-MM-dd`。
- `endDate` 为开始日期加上对应周期后减 1 天。
- `price_cycle` 映射规则：
 - `month` -> `validity=1`
 - `year` -> `validity=2`
 - `day` -> `validity=3`
- `price_cycle_value` 直接作为 `validityUnit`。
- `resource_quotas` 中若不存在 `function` 项, 则回退使用 `feature_switches` 构造 `functionList`。
- 主体信息会结合当前账号的套餐状态、最近销售机会进行推导, 默认主体兜底为广州。

请求示例

```
POST /v3/console/mall-package-version/create-order
Content-Type: application/json

{
  "id": 1001
}
```

返回结构

返回类型：`ResponseVO<PackageRenewalOrderVO>`

ResponseVO 字段

字段名	类型	说明
<code>code</code>	<code>Integer</code>	状态码，成功一般为 <code>200</code>
<code>status</code>	<code>String</code>	<code>success</code> / <code>error</code>
<code>requestId</code>	<code>String</code>	请求 ID
<code>error</code>	<code>Object</code>	失败信息
<code>data</code>	<code>PackageRenewalOrderVO</code>	订单数据

PackageRenewalOrderVO 字段

字段名	类型	说明
<code>orderId</code>	<code>String</code>	订单 ID
<code>tradeNo</code>	<code>String</code>	交易流水号
<code>status</code>	<code>Integer</code>	订单状态
<code>amount</code>	<code>Integer</code>	订单金额，单位分

成功返回示例

```
{
  "code": 200,
  "status": "success",
  "data": {
    "orderId": "202605211830000001",
    "tradeNo": "T202605211830000001",
    "status": 11,
    "amount": 120000
  }
}
```

失败返回示例

```
{
  "code": 400,
  "status": "error",
  "error": {
    "code": 400,
    "desc": "参数错误"
  }
}
```