NOTE:获取订单号，对应的类是：com.u8.server.web.PayAction。可以根据这里进行下单操作。 建议将下单操作，放到游戏服务器进行。

NOTE:2016年12月12号之后的U8SDK版本，下单已经封装在了U8SDK抽象层，无需游戏层进行获取订单号操作了。

说明

游戏中，我们一般都多个充值选项，给用户选择。比如充值10元，获得100元宝；充值50元，获得500元宝等充值选项。当用户在点击该充值选项的时候，我们先去U8Server获取一个订单号。订单号返回的时候，我们才调用渠道SDK的支付接口，打开SDK的支付界面。

为什么要这么做呢？

1、事先获取一个订单号，当渠道SDK异步支付成功，回调到U8Server的时候，我们根据这个订单号来取出订单数据，从而给对应的用户充值。
2、几乎每个渠道SDK都提供了自定义参数。这些自定义参数在支付回调的时候，会原封不动地传给游戏这边。但是，部分渠道这个自定义参数的长度有限，如果我们游戏中有些特殊要求，需要通过自定义参数传一些比较长的参数组合，我们就没办法了。
所以，现在，我们不需要再通过渠道SDK的自定义参数来传递了。我们在获取订单号的时候，也提供了自定义参数。游戏可以把支付成功之后，需要原封不动返回的数据，封装在这个自定义参数中。而U8Server也是游戏公司自己控制，字段的长度可以根据公司自己游戏的业务需求，动态修改。
3、实现订单号，我们会在订单表中生成一个状态为[支付中]的订单，当SDK服务器回调通知的时候，如果成功，将订单状态修改为[支付成功]。紧接着，通知游戏服，如果通知游戏服成功，那么，将订单状态修改为[支付完成]。中间如果支付处理失败，则订单状态修改为[支付失败]
这样，通过后台管理系统，我们可以对订单进行查询，以及对丢单等问题，进行定位。
4、部分渠道SDK，需要自己的下单操作，比如vivo。那么这个时候，当U8Server在下单的时候，要去SDK服务器也进行一次下单操作。当渠道SDK下单返回的时候，将返回的参数，放在extension中，一并返回给游戏客户端。因为客户端可能需要渠道SDK下单返回的数据，才能打开支付界面。

流程

使用U8Server之后，我们的获取订单号流程如下：

1、客户端用户打开游戏内的充值选项界面，点击某个选项
2、游戏客户端发送请求都游戏服务器，获取订单号
3、游戏服务器再去U8Server获取订单号接口进行下单操作，可以携带自定义参数等，在支付成功会原封不动返回的。
4、U8Server在订单表中生成一条订单数据，同时返回当前订单的订单号
5、游戏服务器收到订单号之后，返回给客户端。
6、游戏客户端拿到订单号之后，将其赋值到PayParams数据对象的orderID中，然后调用U8SDK抽象层的U8Pay.getInstance().pay(PayParams data)接口，打开渠道SDK的支付界面。
7、用户在渠道SDK支付界面，进行充值，充值完毕，渠道SDK会返回一个成功或者失败的状态，但是这个成功，并不代表真的支付成功了。仅仅是当前用户支付请求成功。所以，客户端针对这个支付成功，不需要做特殊的操作。可以给用户一个提示，比如"充值成功，到账可能需要延迟几分钟"。
8、渠道SDK收到用户支付请求的时候，处理成功之后，会异步通知到U8Server

接口

NOTE: 如果你还没有搭建好U8Server的开发环境，建议你参考[这篇文档](http://www.6xsdk.com/u8-wiki/#!u8server_setup.md)，先去搭建U8Server的开发环境。

假如你部署好了U8Server，并且根地址是<http://localhost:8080/>

下单地址：http://localhost:8080/pay/getOrderID
请求方式：POST或者GET
参数：
userID：用户登录认证之后u8server生成的userID，每个用户唯一
productID：客户端里面充值选项对应的ID
productName：客户端里面充值选项对应的名称
productDesc：客户端里面充值选项的描述
money：当前充值金额，单位分
roleID：当前游戏内角色ID
roleName：当前游戏内角色名称
roleLevel; 玩家等级
serverID：当前玩家所在的服务器ID
serverName：当前玩家所在的服务器名称
extension：自定义字段，支付回调的时候，会原封不动回调给游戏服务器
notifyUrl：游戏服的支付回调地址，用于u8server支付成功之后，异步通知游戏服务器，游戏服务器给玩家发游戏币或者游戏道具
signType：签名算法，可以是md5或者rsa。 该字段不参与签名
sign：rsa签名（签名算法，见最后）签名使用的私钥是U8Server生成的，创建游戏的时候会生成，给到游戏那边。 该字段不参与签名
返回(json格式)：
{
state: 1（下单成功）；其他失败
data: 认证成功才有数据，否则为空
{
orderID:U8Server生成的唯一订单号，需要传到渠道SDK提供的游戏订单号字段，
或者自定义字段。支付成功回调到U8Server的时候，需要根据该字段来获取订单信息。
extension:当前渠道SDK的扩展数据。比如渠道有下单操作之后，这个字段里面，就存放了渠道SDK下单返回的数据。
}
}