

公告

昵称: Cinlap Soft

园龄: 12年2个月

粉丝: 10

关注: 18

+加关注

最新评论

1. Re: #.NET# DataGrid显示大量数据
——DataGridView虚模式

@StackMiao文章末尾部分已经给出了
如何设置双缓冲的方法...

--Cinlap Soft

2. Re: #.NET# DataGrid显示大量数据
——DataGridView虚模式

怎样使用反射来对表格控件的

DoubleBuffered 属性进行设定? 既然
都讲虚模式了, 何不把详细步骤都讲清楚

--StackMiao

3. Re: CentOS7下安装MariaDB

@段朝阳引用@Cinlap Soft不是切换,
两个没有关系也没有冲突, 基本仓库下载
基本仓库的包, Mariadb下载mariadb
仓库里的软件包。前者有很多常用的软件
可以下载, 比如系统自带的curl之
类.....

--Cinlap Soft

4. Re: CentOS7下安装MariaDB

@Cinlap Soft不是切换, 两个没有关系
也没有冲突, 基本仓库下载基本仓库的
包, Mariadb下载mariadb仓库里的软
件包。前者有很多常用的软件可以下载,
比如系统自带的curl之类的, 后者只
提.....

--段朝阳

5. Re: CentOS7下安装MariaDB

@段朝阳引用@Cinlap Soft你用的阿里
云得源, 只是用了基本仓库的镜像,
mariadb得镜像你没有用。

微信支付 第二篇 JSAPI 调用统一下单接口获取预支付交易数据

上一篇讲到成功获取 **openid**, 本篇要调用微信统一
接口创建预支付交易单

除付款码支付场景以外, 商户系统先调用该接口在
微信支付服务后台生成预支付交易单, 返回正确的
预支付交易会话标识后再按Native、JSAPI、APP
等不同场景生成交易串调起支付。

第三步, 调用微信统一下单接口创建预支付交易单

微信统一下单API是微信支付的一个“统一”处理入口,
官方给出的地址是

```
https://api.mch.weixin.qq.com/pay/unifiedorder
https://api2.mch.weixin.qq.com/pay/unifiedorder(备用域名)
```

首先按照 **统一下单 接口** 中给的 **请求参数** 准备
必要的参数, 在本例 (**JSAPI**) 中至少需要参数如下

- appid, 公众号AppID
- mch_id, 商户号, 一串10位数字
- device_info, 设备号, 非必填但有一定作用,
PC网页或公众号内支付可以传"WEB", 本例中就
设置为"WEB"
- nonce_str, 随机字符串, 请求方提供的随机信
息, 主要保证签名不可预测
- sign, 通过签名算法计算得出的签名值, 详见[签名算法](#)
- sign_type, 签名类型, 截止编辑时支持 (默
认) "MD5"和"HMAC-SHA256", 用哪种算法进
行的签名要通过该字段表示
- body, 商品描述, 商户按规范创建

centosbase.repo和mariadb.repo是两个不相关的仓储linux新人对仓.....

--Cinlap Soft

阅读排行榜

1. DevExpress 学习使用之 LookUpEdit(7045)
2. DevExpress 学习使用之 SplitContainerControl(5017)
3. 咩咩快的 C# 高斯模糊实现(3337)
4. nginx如何设置禁止访问文件或文件夹(2721)
5. 咩咩快的 C# 高斯模糊实现 (续)(2510)

评论排行榜

1. 【C#】聊聊不需要记密码的密码管理(27)
2. Asyncn/Await 异步编程初窥 (二)(11)
3. CentOS7下安装MariaDB(9)
4. C#自动选择出系统中最合适的IP地址(9)
5. #.NET# DataGridView显示大量数据——DataGridView虚模式(2)

- out_trade_no, 商户系统内部订单号, 按微信的规则生成
- total_fee, 整型, 单位分, 订单总金额
- spbill_create_ip, "用户的客户端IP", 这个词儿是微信给出的截止到编辑时从字面上我并不确定指的是请求服务器的服务器地址还是访问商户服务器的某个用户"手机"的地址, 但通过代码应该是后者, 详见之后的代码示例
- notify_url, 异步接收微信支付结果通知的回调地址
- trade_type, 交易类型的标识, JSAPI-JSAPI支付, NATIVE-Native支付, APP-APP支付, MWEB-H5支付
- openid, 第一篇中来自微信用户的openid值

按照 签名算法 计算签名

- 把所有要传递的参数键值对去掉值是空的, 剩下的参数名ASCII码从小到大排序后, 使用URL键值对格式 (key1=value1&key2=value2...) 拼接成StringA
- 把StringA后面多加一组键值 **&key=商户平台密钥** 获得StringB
- 把StringB按要求做"MD5"或"HMAC-SHA256"计算, 并将结果字符转为大写
- 微信甚至还提供了一个在线校验工具帮助开发者检查生成的签名是否正确, [跳到工具](#)。使用方法是选择好 **签名类型**, 校验方式选择 **XML** (不是必须只是为了省事儿), **XML源串** 输入不带 **sign** 信息的部分如

```
<xml>
  <appid>wxcx5bbf</appid>
  <mch_id>15xxxxxxxxxx61</mch_id>
  <device_info>WEB</device_info>

  <nonce_str>6743482E92E61DE587B52D6C7DEBFE79<
/nonce_str>
  <body>xx科技-测试</body>
  <out_trade_no>TY-OUTTRADENO-TEST-
1561440044</out_trade_no>
  <total_fee>1</total_fee>

  <spbill_create_ip>36.xx.xx.185</spbill_creat
e_ip>
```

```
<notify_url>http://wxpay.txxx.com/jsapi/unifiedOrderNotify.php</notify_url>
    <trade_type>JSAPI</trade_type>
    <openid>100003</openid>
</xml>
```

再输入正确的 **商户密钥**，点击按钮即可计算出结果，与自己计算的结果进行比对。

- 本例中 **Sign** 方法接收两个参数，**\$arrayobj** 为数组类型，保存了所有参数名和参数值组成的字符串数组，方法输出计算后的签名结果。
ToUrlParams 方法将参数数组拼接为URL格式字符串。

```
/**
 * 生成签名
 * @param 参与签名的内容组成的数组
 * @param 是否使用 HMAC-SHA256算法，否则使用 MD5
 * @return 签名字符串
 * @author 试试手气
 * @version test pass
 */
public static function Sign($arrayobj, $hmacsha256 = true)
{
    //步骤一：字典排序
    ksort($arrayobj);

    //步骤二：在
    $str = Util::ToUrlParams($arrayobj);

    //步骤三：在$str后面加KEY
    $mchKey = "asdxxxxxxxxxxxxxN82"; // 商户支付密钥，生产代码中是不可能如此硬编码到这里的
    $str .= "&key=" . $mchKey;

    //步骤四：MD5或HMAC-SHA256加密
    if ($hmacsha256) {
        $str = hash_hmac("sha256", $str, $mchKey);
    } else {
        $str = md5($str);
    }

    //步骤五：所有字符转大写
    $result = strtoupper($str);
    return $result;
}
```

```

        public static function
        ToUrlParams($arrayobj)
        {
            $buff = "";
            foreach ($arrayobj as $k => $v) {
                if ($k != "sign" && $v != "" &&
                !is_array($v)) {
                    $buff .= $k . "=" . $v .
                    "&";
                }
            }

            $buff = trim($buff, "&");
            return $buff;
        }
    
```

将签名结果作为最后的参数一起创建XML格式的POST数据

除 **sign** 签名外，其它的参数要么是现成的，要么是可以运行时获取到的如 **spbill_create_ip**。计算完签名，所有的参数就都准备好了，把它们加工成XML，参数顺序没有提到有限制。

```

function toXML($paramsobj)
{
    $xml = "<xml>\n";

    foreach ($paramsobj as $key => $value) {
        $xml .= "<" . $key . ">" . $value .
        "</" . $key . ">\n";
    }

    $xml .= "</xml>";
    return $xml;
}
    
```

出来是这个样子的

```

<xml>
    <appid>wxcxxxxx5bbf</appid>
    <mch_id>15xxxxxxxxxxxxx61</mch_id>
    <device_info>WEB</device_info>

    <nonce_str>6743482E92E61DE587B52D6C7DEBFE79<
    /nonce_str>
    <body>xx科技-测试</body>
    <out_trade_no>TY-OUTTRADENO-TEST-
    1561440044</out_trade_no>
    <total_fee>1</total_fee>
    
```

```
<spbill_create_ip>36.xx.xx.185</spbill_create_ip>

<notify_url>http://wxpay.txxx.com/jsapi/unifiedOrderNotify.php</notify_url>
    <trade_type>JSAPI</trade_type>
    <openid>100003</openid>

<sign>42DDF6C558EC05E24AAD0Cxxxxxxxx1DC3A53C0053D4A6F42E31036132F74481</sign>
</xml>
```

向微信统一下单API提交XML

用POST方式向统一下单接口地址提交数据，本例中使用的是微信SDK中的 **postXmlCurl** 方法改造了一下，用的是 **curl** 访问网络

```
function postXmlCurl($config, $xml, $url,
    $useCert = false, $second = 30)
{
    $ch = curl_init();
    //$curlVersion = curl_version();
    //$ua = "WXPAYSDK/" . self::$VERSION .
    (" . PHP_OS . ") PHP/" . PHP_VERSION .
    CURL/" . $curlVersion['version'] . " "
    //      . $config->GetMerchantId();

    //设置超时
    curl_setopt($ch, CURLOPT_TIMEOUT,
    $second);

    //      $proxyHost = "0.0.0.0";
    //      $proxyPort = 0;
    //      $config->GetProxy($proxyHost,
    $proxyPort);
    //      //如果有配置代理这里就设置代理
    //      if($proxyHost != "0.0.0.0" &&
    $proxyPort != 0){
    //          curl_setopt($ch, CURLOPT_PROXY,
    $proxyHost);
    //          curl_setopt($ch, CURLOPT_PROXYPORT,
    $proxyPort);
    //      }
    curl_setopt($ch, CURLOPT_URL, $url);

    //curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, TRUE);

    //curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
    //严格校验
```

```

//curl_setopt($ch,CURLOPT_USERAGENT,
$ua);
//设置header
curl_setopt($ch, CURLOPT_HEADER, FALSE);
//要求结果为字符串且输出到屏幕上
curl_setopt($ch, CURLOPT_RETURNTRANSFER,
TRUE);

if ($useCert == true) {
    //设置证书
    //使用证书: cert 与 key 分别属于两个.pem
文件
    //证书文件请放入服务器的非web目录下
    $sslCertPath = "";
    $sslKeyPath = "";
    $config-
>GetSSLCertPath($sslCertPath, $sslKeyPath);
    curl_setopt($ch,
CURLOPT_SSLCERTTYPE, 'PEM');
    curl_setopt($ch, CURLOPT_SSLCERT,
$sslCertPath);
    curl_setopt($ch, CURLOPT_SSLKEYTYPE,
'PEM');
    curl_setopt($ch, CURLOPT_SSLKEY,
$sslKeyPath);
}
//试试手气新增, 增加之后 curl 不报 60# 错误,
可以请求到微信的响应
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER,
false); //不验证 SSL 证书
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST,
false); //不验证 SSL 证书域名
    //post提交方式
    curl_setopt($ch, CURLOPT_POST, TRUE);
    curl_setopt($ch, CURLOPT_POSTFIELDS,
$xml);
    //运行curl
    $data = curl_exec($ch);
    //返回结果
    if ($data) {
        curl_close($ch);
        return $data;
    } else {
        $error = curl_errno($ch);
        curl_close($ch);
        return "curl error, error code
".$error;
        //throw new WxPayException("curl出
错, 错误码:$error");
    }
}

```

没有删掉代码，不用的注释掉了，并增加了关于 **CURLOPT_SSL_VERIFYPEER** 和 **CURLOPT_SSL_VERIFYHOST** 的明确设置，在增加这两句之前curl本身会报60错误。

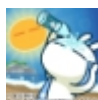
返回值

按微信文档，访问（非业务）成功将返回（至少）包含 **return_code** 和 **return_msg** 两个字段的XML文档，对照统一下单接口文档内的返回结果即可判断请求是否成功，如下例

```
<xml>
  <return_code><![CDATA[SUCCESS]]>
</return_code>
  <return_msg><![CDATA[OK]]></return_msg>
  <appid><![CDATA[wxcccxxxbbf]]></appid>
  <mch_id><![CDATA[15xxx61]]></mch_id>
  <device_info><![CDATA[WEB]]>
</device_info>
  <nonce_str><![CDATA[SMIpZtHhPPyTezlY]]>
</nonce_str>
  <sign><![
[CDATA[7215B1BAB1C08A4887AAB0ECB2A6044729FE8
A8BC6E70591660A6BCC11D7DC9A]]></sign>
  <result_code><![CDATA[SUCCESS]]>
</result_code>
  <prepay_id><![
[CDATA[wx251xxxxxxxxxxxxxxxxxxxxxxxxxxxx078
700]]></prepay_id>
  <trade_type><![CDATA[JSAPI]]>
</trade_type>
</xml>
```

很明显是访问也成功了，业务也成功了，也就是接口业务成功生成了 **预支付交易会话 wx251xxxxxxxxxxxxxxxxxxxxxxxxxxxx 078700**，后面就该使用该会话标识调起H5支付了。

不重要的其实最重要

[好文要顶](#)[关注我](#)[收藏该文](#)

Cinlap Soft

关注 - 18

粉丝 - 10

[+加关注](#)

0

0

« 上一篇: [微信支付 第一篇 JSAPI 支付配置与获取 OpenID](#)

» 下一篇: [微信支付 第三篇 微信调用H5页面进行支付](#)

posted @ 2019-06-25 17:03 Cinlap Soft 阅读

(133) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【推荐】ALIYUN90%，大陆-港澳专线直连，创业者上云首选

【推荐】程序员问答平台，解决您开发中遇到的技术难题

相关博文:

- [微信小程序支付及退款流程详解](#)
- [pringboot整合微信支付sdk](#)
- [SpringBoot项目后台对接微信支付开发——微信统一下单接口开发](#)
- [微信JSAPI支付 统一下单 \(.net后台\)](#)
- [微信支付 jsapi接口](#)