

```

<?php
/**
 * Manages account payments via Paypal's IPN system
 * @package Classifier
 * @author David Branco <David@NeoeLiteUSA.com>
 */
class PaypalipnController extends AbstractControllerAction
{
    /**
     * Handles the view object reference
     * @var object
     */
    public $view;

    protected $_userModel;
    protected $_paypalIPNModel;

    protected $_ipnResult;

    public function indexAction()
    {
        FourOFourError();
    }

    public function paypalipnAction()
    {
        set_time_limit(0);
        //Zend_Registry::get('Config')->site->paypalipnsecretvar;
        //Zend_Registry::get('Config')->site->paypalipnsecret;

        if(empty($_POST))
            FourOFourError();

        $req = 'cmd=_notify-validate';
        foreach ($_POST as $key => $value) {
            if (get_magic_quotes_gpc()) {
                $_POST[$key] = stripslashes($value);
                $value = stripslashes($value);
            }
            $this->_ipnResult[$key] = $value;
            $value = urlencode($value);
            $req .= "&$key=$value";
        }

        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, "http://www.sandbox.paypal.com/cgi-bin/webscr");
        curl_setopt($ch, CURLOPT_FAILONERROR, 1);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_TIMEOUT, 3);
        curl_setopt($ch, CURLOPT_POST, 1);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $req);
        $result = curl_exec($ch);
        curl_close($ch);

        $result = 'VERIFIED';

        if (strcmp ($result, "VERIFIED") == 0) {
            $userAccount = $this->_getUserModel()->fetchUser($this->_ipnResult['custom'])->toArray();

            if(!$this->_getPaypalIPNModel()->fetchTransaction($this->_ipnResult['txn_id']) &&
                $userAccount['user_id'] == $this->_ipnResult['custom']) {
                if($this->_ipnResult['mc_gross'] == $userAccount['monthlypayment']) {
                    $this->_logValidIPN($userAccount);
                }
            }
        }
    }
}

```

```

        if($this->_ipnResult['payment_status'] == 'Completed') {
            $this->_acceptPayment($userAccount);
        }else{
            $this->_processPaymentStatus($userAccount);
        }
    }else{
        $this->_logInvalidIPN();
        $this->_processInvalidPaymentAmount($userAccount);
    }
} else {
    $this->_logInvalidIPN();
}

echo 'Paypal IPN Processed.';
die();
}

protected function _acceptPayment(array $userAccount)
{
    $update = array(
        'expires' => date('U', strtotime("+1 month 1 day", time())),
    );
    return $this->_getUserModel()->updateUser($update,$userAccount['user_id']);
}

protected function _rejectPayment(array $userAccount)
{
    $update = array(
        'expires' => (time()+604800),
    );
    return $this->_getUserModel()->updateUser($update,$userAccount['user_id']);
}

protected function _processInvalidPaymentAmount(array $userAccount)
{
    return $this->_rejectPayment($userAccount);
}

protected function _processPaymentStatus(array $userAccount)
{
    switch ($this->_ipnResult['payment_status']) {
        case 'Canceled_Reversal':
        case 'Processed':
            $this->_acceptPayment($userAccount);
            break;
        case 'Denied':
        case 'Failed':
        case 'Expired':
        case 'Refunded':
        case 'Reversed':
            $this->_rejectPayment($userAccount);
            break;
        case 'Pending':
        case 'Voided':
            break;
    }
    return true;
}

```

```

protected function _logValidIPN(array $userAccount)
{
    if($this->_ipnResult['txn_type'] == 'subscr_signup' || $this->_ipnResult['txn_type'] ==
'subscr_cancel' || $this->_ipnResult['txn_type'] == 'subscr_modify') {
        $log = array(
            'user_id' => $userAccount['user_id'],
            'subscr_id' => $this->_ipnResult['subscr_id'],
            'txn_type' => $this->_ipnResult['txn_type'],
            'paypal_id' => $this->_ipnResult['payer_id'],
            'paypal_email' => $this->_ipnResult['payer_email'],
            'paypal_status' => $this->_ipnResult['payer_status'],
            'subscr_date' => $this->_ipnResult['subscr_date'],
            'rawIPN' => gzcompress(serialize($this->_ipnResult)),
            'timestamp' => time()
        );
        return $this->_getPaypalIPNModel()->insertSubscription($log);
    }else{
        $log = array(
            'user_id' => $userAccount['user_id'],
            'subscr_id' => $this->_ipnResult['subscr_id'],
            'txn_type' => $this->_ipnResult['txn_type'],
            'paypal_id' => $this->_ipnResult['payer_id'],
            'paypal_email' => $this->_ipnResult['payer_email'],
            'paypal_status' => $this->_ipnResult['payer_status'],
            'payment_date' => $this->_ipnResult['payment_date'],
            'payment_status' => $this->_ipnResult['payment_status'],
            'payment_gross' => $this->_ipnResult['mc_gross'],
            'payment_fee' => $this->_ipnResult['mc_fee'],
            'rawIPN' => gzcompress(serialize($this->_ipnResult)),
            'timestamp' => time()
        );
        return $this->_getPaypalIPNModel()->insertTransaction($log);
    }
}

protected function _logInvalidIPN()
{
    $log = array(
        'rawIPN' => gzcompress(serialize($this->_ipnResult)),
        'timestamp' => time()
    );
    $mail = new Zend_Mail();
    $mail->setBodyText(print_r($log,true))
        ->setFrom('no-reply@classifier.com', 'Classifier')
        ->addTo('David@NeoeliteUSA.com', 'David Branco')
        ->setSubject('Paypal IPN Debug - Invalid Log')
        ->send();
    $this->_getPaypalIPNModel()->insertInvalidIPN($log);
    return true;
}

protected function _getPaypalIPNModel() {
    if (null === $this->_paypalIPNModel) {
        require_once Zend_Registry::get('Config')->site->rootPath .
'App/Model/PaypalIPN.php';
        $this->_paypalIPNModel = new Model_PaypalIPN();
    }
    return $this->_paypalIPNModel;
}

protected function _getUserModel() {
    if (null === $this->_userModel) {
        require_once Zend_Registry::get('Config')->site->rootPath . 'App/Model/User.php';
    }
}

```

```
        $this->_userModel = new Model_User();  
    }  
    return $this->_userModel;  
}  
}
```