

(/)



Perl New Features: A book about the new features in Perl up to Perl v5.42 [Learn more](#)

MOCK / Business-OnlinePayment-StoredTransaction-0.01 / lib / Business / OnlinePayment / StoredTransaction.pm

```
1 package Business(/)::OnlinePayment::StoredTransaction;
2
3 use 5.008004;
4 use strict;
5 use warnings;
6 use Carp;
7 use Business::OnlinePayment;
8 use Crypt::OpenSSL::RSA;
9 use Crypt::CBC;
10 use Storable;
11 use MIME::Base64;
12 use Digest::MD5;
13
14 our @ISA = qw(Business::OnlinePayment);
15
16 our @EXPORT_OK = ();
17
18 our @EXPORT = ();
19
20 our $VERSION = '0.01';
21
22
23 # Preloaded methods go here.
24
25 sub set_defaults {
26     my $self = shift;
27     $self->build_subs(qw(public_key));
28 }
29
30 sub map_fields {
31     my $self = shift;
32     my %content = $self->content();
33     $content{'type'} = lc($content{'type'});
34     $content{'action'} = lc($content{'action'});
35     $self->transaction_type($content{'type'});
36     $self->public_key($content{'password'});
37     $content{'password'} = '';
38     $content{'name'} = $content{'first_name'}.' '.$content{'last_name'}
39     unless defined $content{'name'};
40     $content{'expiration'} =~ /(\d\d)\D*(\d\d)/ if $content{'expiration'};
41     $content{'expiration_month'} = $1
42     unless defined $content{'expiration_month'};
43     $content{'expiration_year'} = $2
44     unless defined $content{'expiration_year'};
45     $content{'currency'} = 'USD $'
46     unless defined $content{'currency'};
47     $self->content(%content);
48 }
49
50 sub submit {
51     my $self = shift;
52     my %actions = ('normal authorization' => 1,
53                   'authorization only' => 1,
54                   'credit' => 1,
55                   'post authorization' => 1,
56                   'void' => 1);
```

```

57 $self->map_fields();
58 my $content = $self->content();
59 my $public_key = $self->public_key();
60 croak "No public key found in 'password'" unless $public_key;
61 if ($actions{$content{action}}) {
62     my $rsa_pub = Crypt::OpenSSL::RSA->new_public_key($public_key);
63     my $plaintext = Storable::nfreeze($content);
64     my $seckey = Digest::MD5::md5_hex(rand());
65     my $encseckey;
66     eval { $encseckey = $rsa_pub->encrypt($seckey) };
67     my $cipher = Crypt::CBC->new( {'key' => $seckey,
68                                   'cipher' => 'Blowfish',
69                                   });
70     my $ciphertext = $cipher->encrypt($plaintext);
71     $ciphertext = encode_base64($ciphertext);
72     $ciphertext =~ s/\s+//g;
73     $encseckey = encode_base64($encseckey);
74     $encseckey =~ s/\s+//g;
75     $ciphertext = "$encseckey:$ciphertext";
76     if ($ciphertext and !$@) {
77         $self->is_success(1);
78         $self->authorization($ciphertext);
79         $self->error_message("success");
80     }
81     else {
82         $self->is_success(0);
83         $self->error_message("failed to encrypt @$");
84     }
85 }
86 else {
87     croak "Bad Action >$content{action}< - That action is not supported"
88 }
89 }
90
91 1;
92 __END__
- Show 76 lines of Pod

```