

PHP Issues Solution

Content

AuthorizeNet Repository Instructions	4
Issue 1 - 462 - Escaping JSON values	21
Cause and solution:	21
Evidence	22
Issue 5 - 446 - Payment decline but receive payment in my authorize wallet.....	28
Cause and solution:	28
Issue 10 - 432 - After upgrading from 1.9.9 to ^2.x having "className" not found during the class mapping.....	30
Solution:	32
Evidence	33
Issue 11 – 430 – CreateTransitionController returns null, conflicting examples.....	34
Cause and Solution:.....	34
Issues 13 - 426 - E00007: User authentication failed due to invalid authentication values	35
Cause and Solution:.....	35
Issue 17 – 421 – how to send extra options in charge Customer Profile? I want to send x duplicate Windows = 0 in my function which is charging saved profile	37
Cause:.....	37
Solution:	37
Issue 18 – 418 - getTransId() not returning expected result.....	39
Cause and solution:	39
Issue 24 – 411 - Other payment types are not supported?	41
AutotizeNet Payment Methods	41
Payment Method – Debit (eCheck).....	41
Payment method – Credit	57
Get transaction key and ID.....	61
Evidence of proof	68
Debit.....	68
Transaction message.....	69
Credit.....	69
Transaction message.....	70
Verify transactions	70
Issue 26 – 409 - Error 404.....	73
Cause:.....	73
Solution:	73

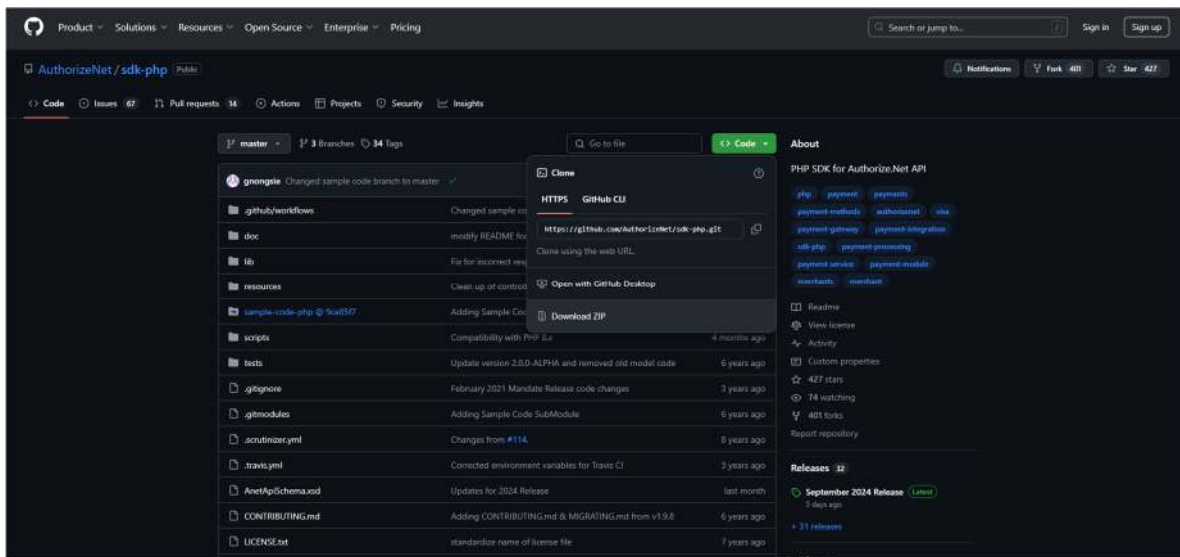
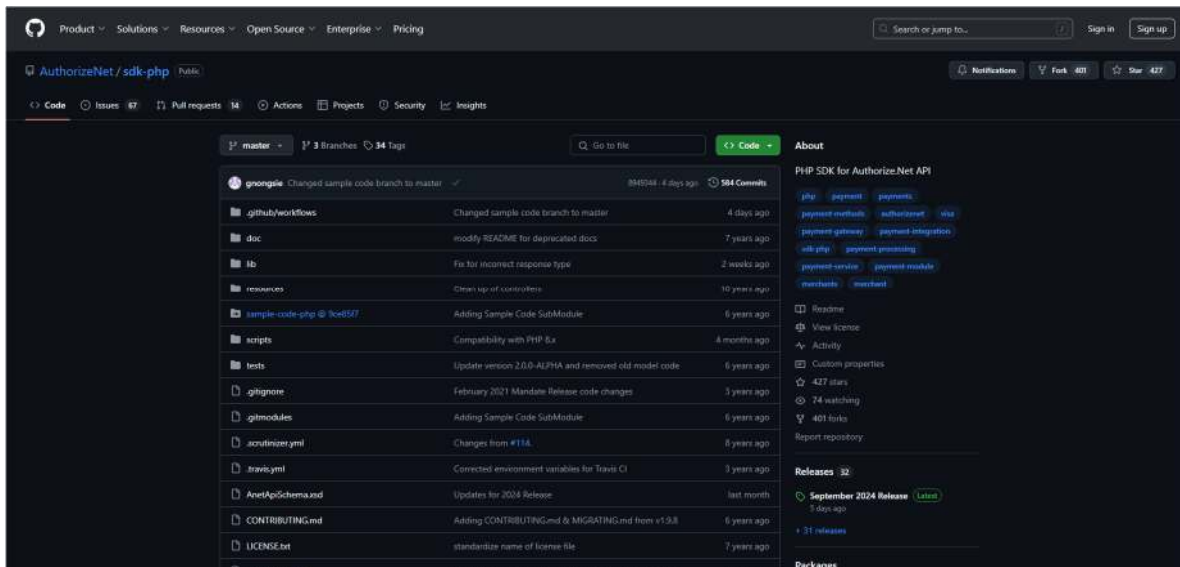
Issue 27 – 405 - Get Customer Profile with merchantCustomerID	74
Cause and solution:	74
Issue 28 – 404 - invalid space characer in GetTransactionDetailsResponse.php on line 96.....	75
Cause:	75
Solution:	75
Issue 29 – 403 - How do you set taxes?	77
Cause:	77
Solution:	77
Issue 30 – 402 - Is there better documentation?.....	79
Cause and solution:	79
Issue 32 - 398 – [Blocker] Composer Update giving error	81
Cause and solution:	81
Evidence	82

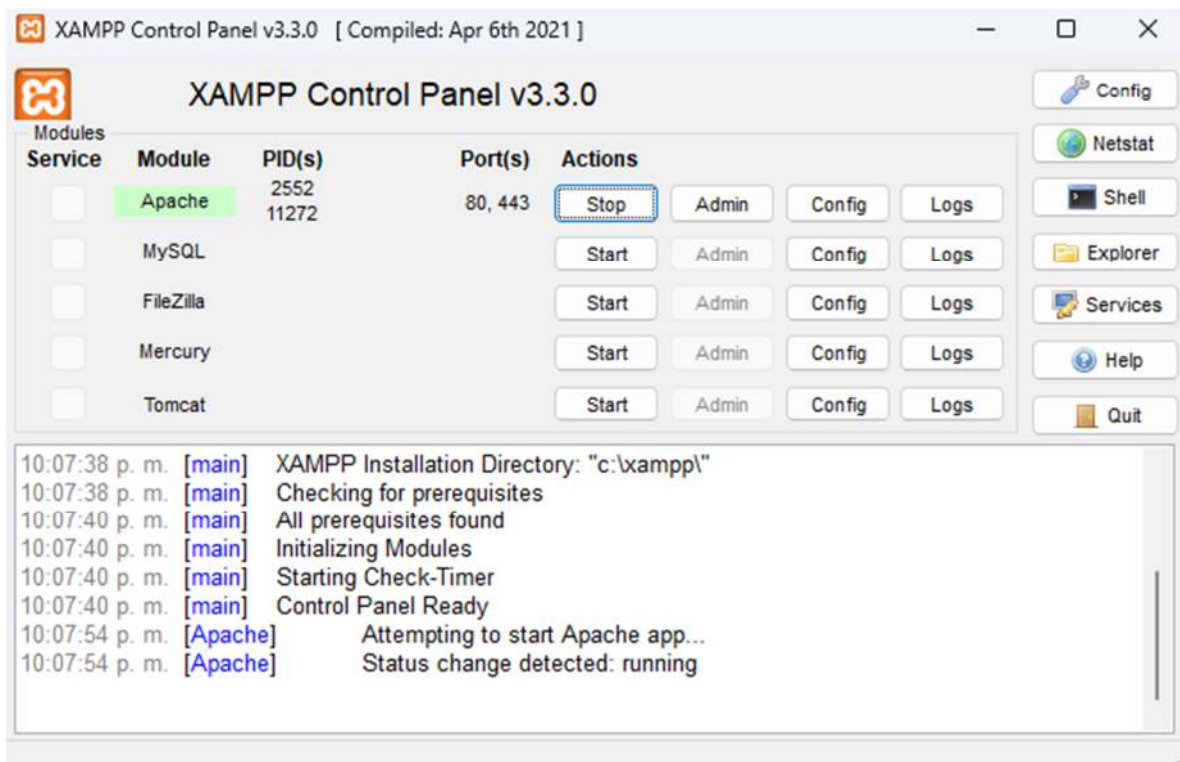
AuthorizeNet Repository Instructions

Download AuthorizeNet repository – SDK-PHP

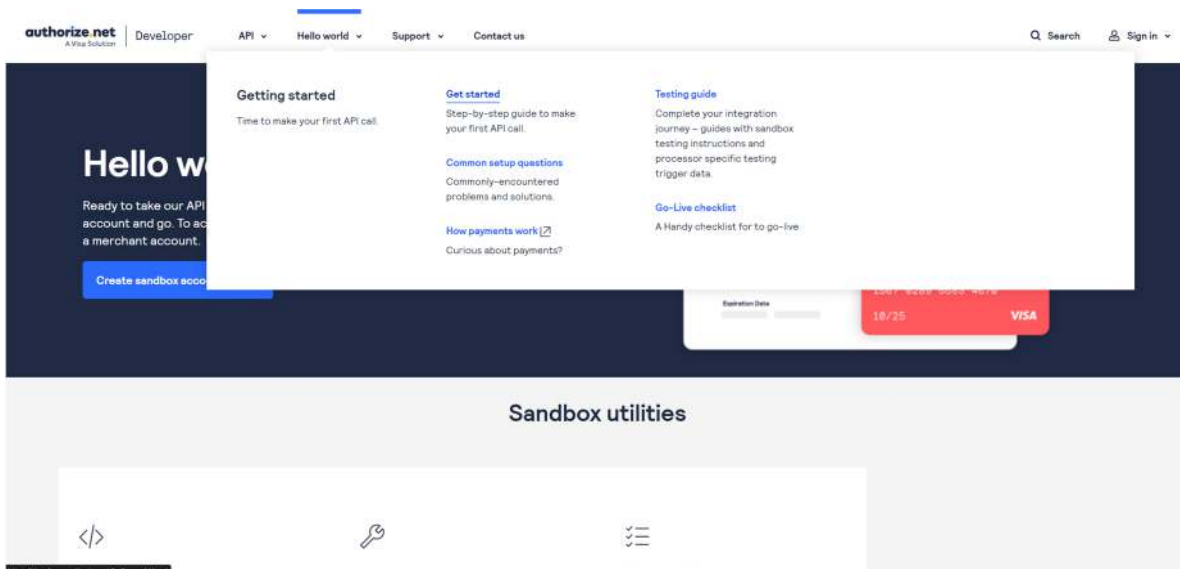
Link:

<https://github.com/AuthorizeNet/sdk-php>





On the same authorize.net page I followed the instructions to generate the file that we will call payments that contains all the essential data to send the payment data and that will communicate with the SDK files to generate the payment.



We select the PHP option

Step 1

Get the Authorize.net Java sample code

Our Java sample code project provides samples of all our API features & functions. Get the [sample code project](#) from GitHub. If you are adding the SDK to an existing project simply update your POM file. For example:

```
1 <groupId>net.authorize</groupId>
2 <artifactId>anet-java-sdk</artifactId>
3 <version>LATEST</version>
```

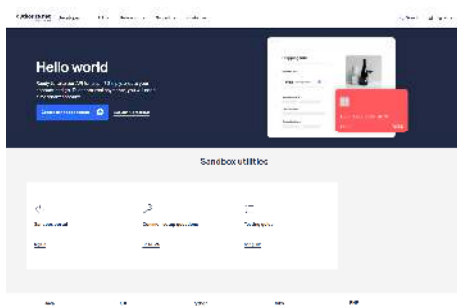


Step 2

Charge a credit card

The Hello World program for payments is charging a credit card. The code below ([source code](#)) will charge a test card and print the auth code & transaction ID to

The instructions explain all the steps to follow to configure the SDK, the name of the file to be generated is optional for the developer.



Step 1

Import the Kotlin library

Next, we need to import the Kotlin library into our Java code. This is done by adding the following import statement at the top of the file:

```
import org.jetbrains.kotlin.jvm.*
```

```
1 // Import the Kotlin library
2 import org.jetbrains.kotlin.jvm.*
3
4 // Main function
5 fun main() {
6     // Print "Hello World"
7     println("Hello World")
8 }
```

Step 2

Compile and Run the Application

To compile and run the application, we need to use the Kotlin compiler. We can do this by running the following command in the terminal:

```
1 // Compile and Run the Application
2
3 // Compile the application
4 kotlin Main.kt
5
6 // Run the application
7 java -cp . Main
8
9 // Output: Hello World
10
11 // Explanation:
12 // The Kotlin compiler (kotlin) is used to compile the Main.kt file into a Java class file (Main.class).
13 // The Java Virtual Machine (JVM) is then used to run the Main.class file.
14 // The output of the application is "Hello World".
15
16 // Additional Information:
17 // - The Kotlin compiler (kotlin) is a command-line tool that can be installed globally or locally.
18 // - The Java Virtual Machine (JVM) is a software layer that enables applications to run on a computer.
19 // - The output of the application is "Hello World".
20
21 // Conclusion:
22 // This tutorial has shown how to import the Kotlin library, compile the application, and run the application.
23 // The output of the application is "Hello World".
```

Step 3

Run

Run the application by clicking the Run button in the IDE.



In this case, some modifications were made to make the display and operation more intuitive for the user; this was only done with the developer's test functionality.

Annex code and document

```
<?php
require 'vendor/autoload.php';
use net\authorize\api\contract\v1 as AnetAPI;use net\authorize\api\controller as AnetController;

// *****
// *****
record our payment data first$amount = $_POST['amount'];$name = $_POST['name'];$street =
$_POST['street'];$city = $_POST['city'];$state = $_POST['state'];$zip = $_POST['zip'];$card =
$_POST['card'];$card_exp_month = $_POST['card_exp_month'];$card_exp_year =
$_POST['card_exp_year'];$ex_date = $card_exp_year.'-'. $card_exp_month;$cvv = $_POST['
cvv'];$email = $_POST['email'];// *****
// *****
*****

define("AUTHORIZENET_LOG_FILE","phplog");

// Common setup for API credentials $merchantAuthentication = new
AnetAPI\MerchantAuthenticationType(); $merchantAuthentication->setName("4U293qwGz");
$merchantAuthentication->setTransactionKey("3vS6nvUS78zCi23c "); $refId = 'ref' . time();

// *****
// *****
Create the payment data for a credit card $creditCard = new AnetAPI\CreditCardType();
$creditCard->setCardNumber($card); $creditCard->setExpirationDate($ex_date); $creditCard-
>setCardCode("127"); // add in $paymentOne = new AnetAPI\PaymentType(); $paymentOne-
>setCreditCard($creditCard);

// Set the customer's identifying information
$customerData = new AnetAPI\CustomerDataType(); $customerData->setType("individual");
$customerData->setEmail($email); // Order info $order = new AnetAPI\OrderType(); $order-
>setInvoiceNumber("698712"); $order->setDescription("Test Product");

// Set the customer's Bill To address add this section in $customerAddress = new
AnetAPI\CustomerAddressType(); $customerAddress->setFirstName($name); $customerAddress-
>setAddress($street); $customerAddress->setCity($city); $customerAddress->setState($state);
$customerAddress->setZip($zip); $customerAddress->setCountry("USA"); // end of customer
billing info code

// Create a transaction $transactionRequestType = new AnetAPI\TransactionRequestType();
$transactionRequestType->setTransactionType("authCaptureTransaction");
$transactionRequestType->setAmount($amount); $transactionRequestType->setOrder($order); //
add in $transactionRequestType->setCustomer($customerData); // add in
```

```

$transactionRequestType->setBillTo($customerAddress); // add in $transactionRequestType-
>setPayment($paymentOne); $request = new AnetAPI\CreateTransactionRequest(); $request-
>setMerchantAuthentication($merchantAuthentication); $request->setRefId($refId); $request-
>setTransactionRequest($transactionRequestType); $controller = new
AnetController\CreateTransactionController($request); $response = $controller-
>executeWithApiResponse(\net\authorize\api\constants\ANetEnvironment::SANDBOX); //
*****
*****
*****

if ($response != null){ $tresponse = $response->getTransactionResponse(); if (($tresponse != null)
&& ($tresponse->getResponseCode()=="1")) { header("Location: ./?m=1"); die(); } else {
header("Location: ./?m=2"); die(); }}
else{ header("Location: ./?m=2"); die();}?>

```

In the code, the highlighted and underlined parts are the data to be modified, which I will explain below:

require 'vendor/autoload.php';

This line requires that you mention the location of the autoload.php file which is in charge of processing the data from the file we named as payment and directing them to their respective algorithms, classes and functions that the SDK contains, by default it is located in the following link vendor->autoload.php, in case you modify the location of this file, you must modify this line of code to its exact location.

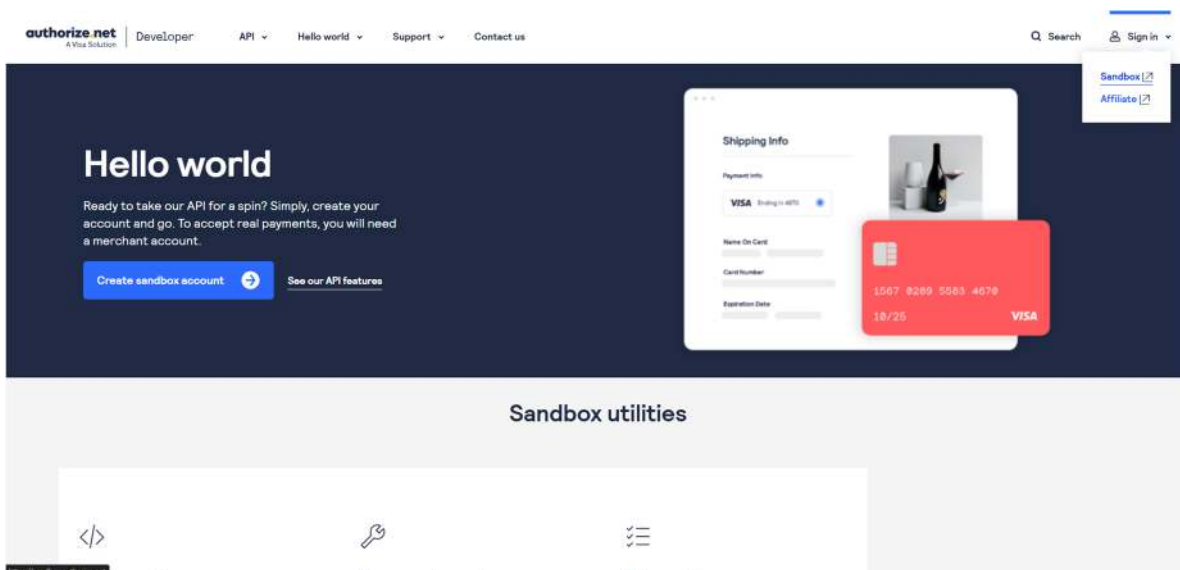
```

$merchantAuthentication->setName("4U293qwGz"); $merchantAuthentication-
>setTransactionKey("3vS6nvUS78zCj23c ");

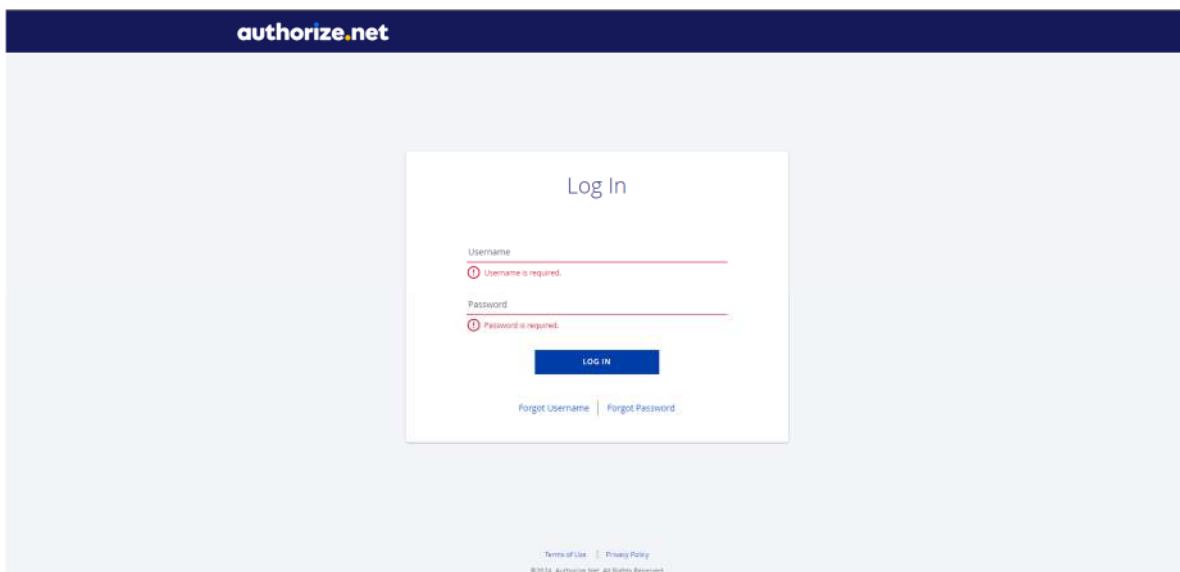
```

These lines of code require the user's personal data in the sandbox, which are found as follows:

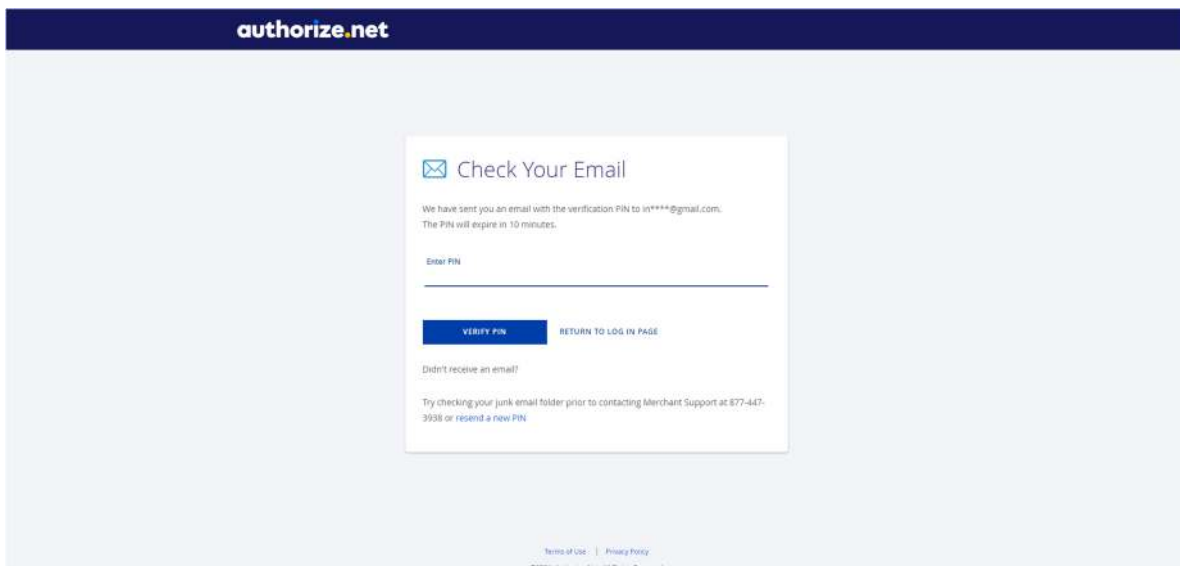
1. We enter the AuthorizeNet page and log in with the Sandbox account



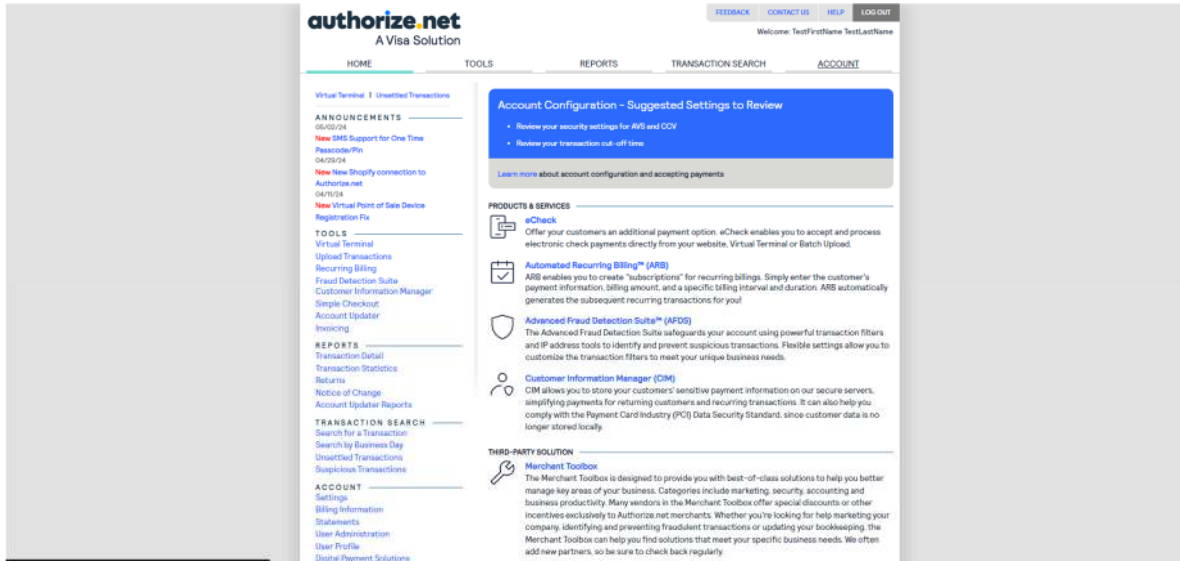
2. We enter our credentials



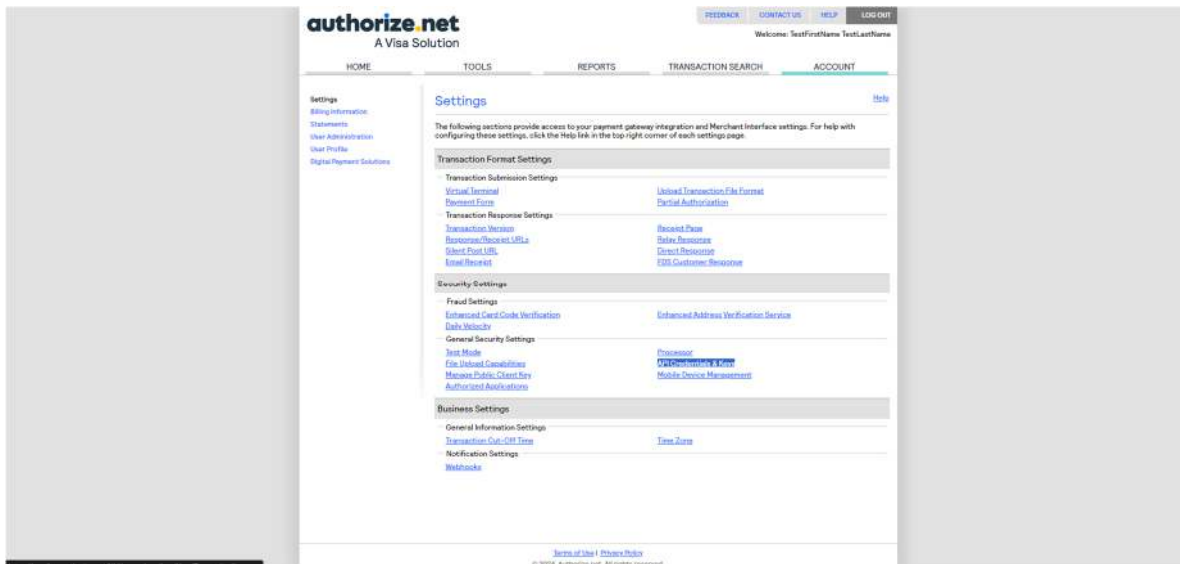
3. We enter the confirmation code that was sent to our email



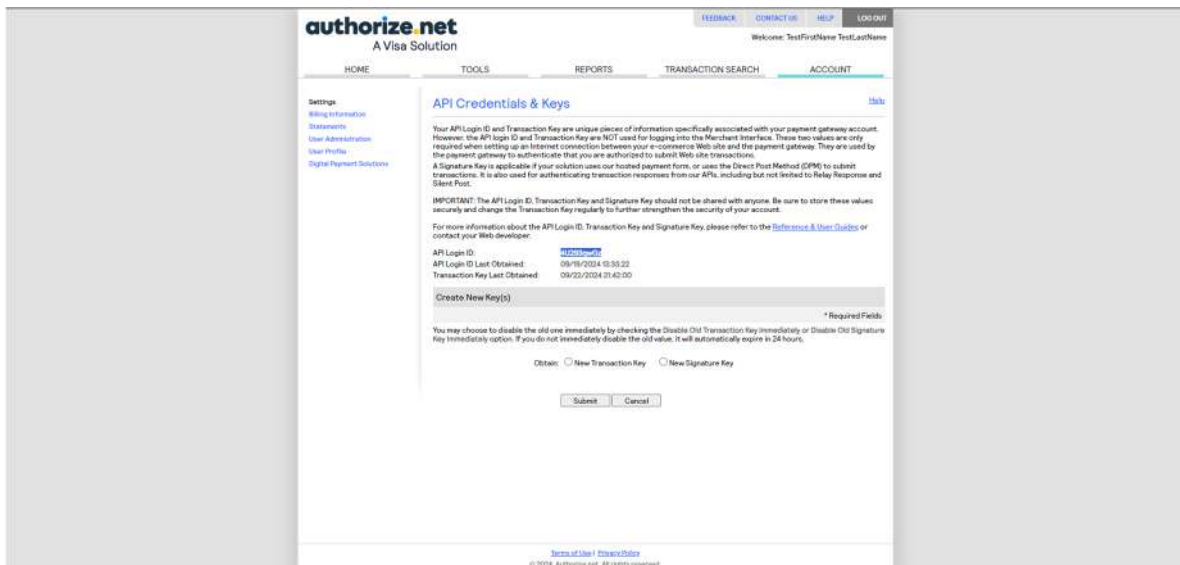
4. Go to the Account option



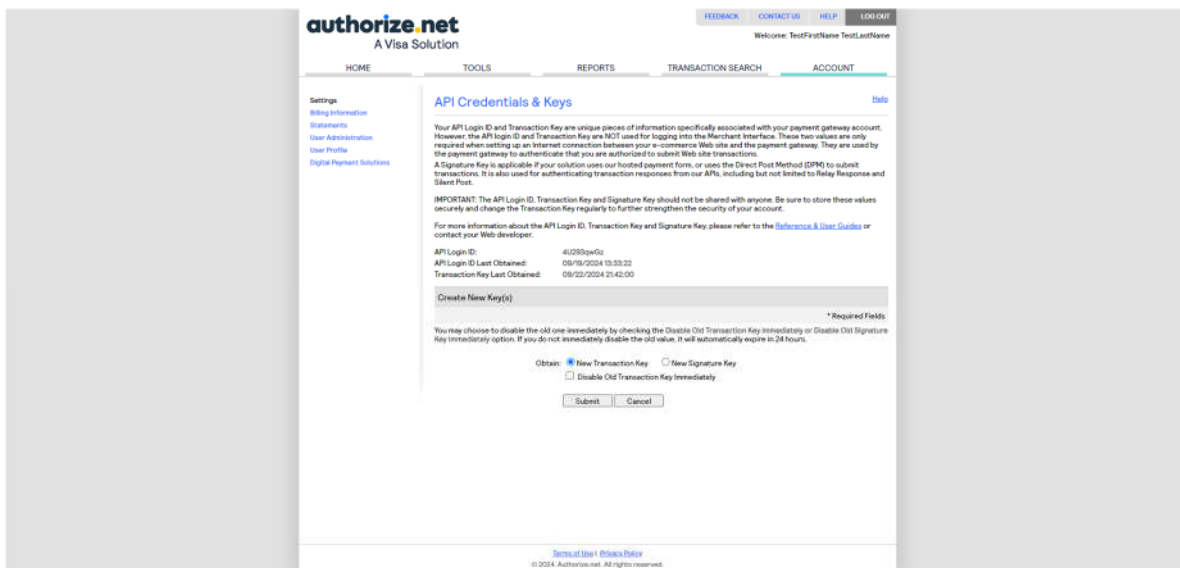
5. In the Security Settings → General Security Settings option select the API Credentials & Keys option



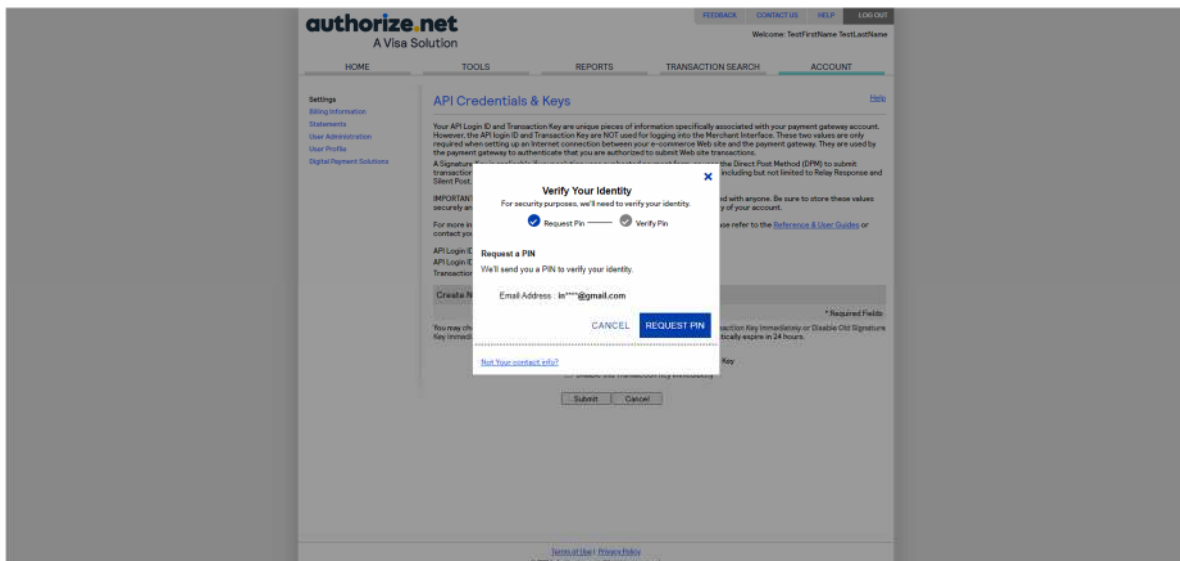
6. In this option you will find the user ID that we must copy and paste into this line of code
`$merchantAuthentication->setName("4U293qwGz");`



7. To obtain the Transaction Key we must perform the following steps: Create new Key(s)

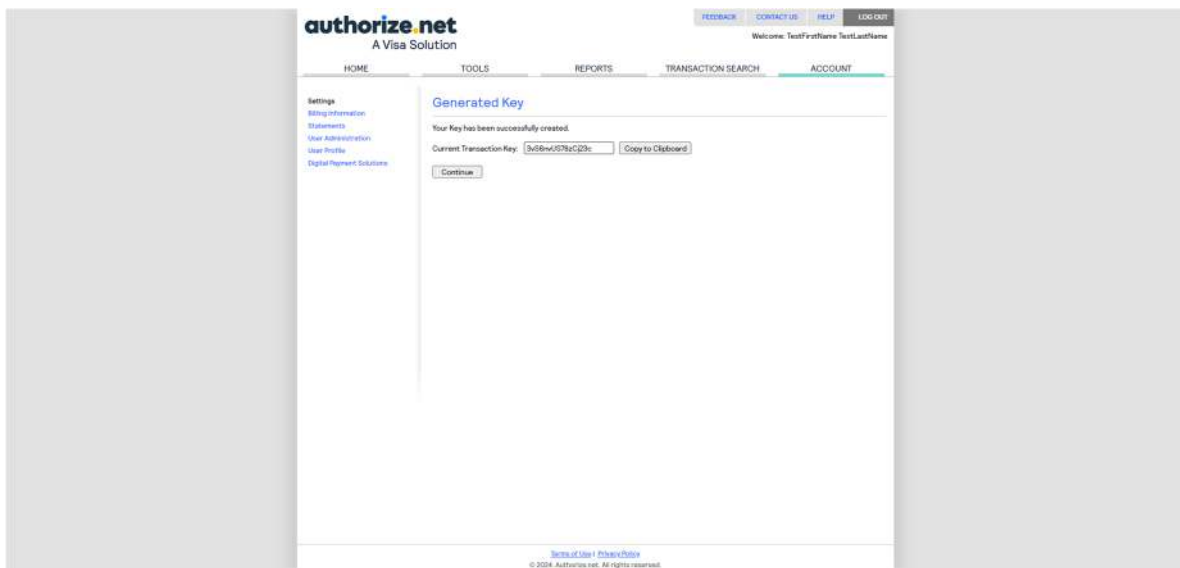


8. We will have the option to disable the previous key. It will also ask us to verify with a PIN via email to show us the data.



9. Once verified, we will be able to view the transaction key that we must copy in the following line:

```
$merchantAuthentication->setTransactionKey("3vS6nvUS78zCj23c");
```



Once all these configurations have been made, we must create an Index file to display the user interface. The following code is provided by the developer and only meets the basic visual characteristics for carrying out the test in our case.

```
<?php$m = $_GET['m'];
```

```
?><!doctype html><html lang="en"> <head>
```

```
<meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no"> <title>Check Out</title>
```

```
<style>
```

```
body { font-family: Arial Narrow, sans-serif;}.content-wrapper { width: 400px; margin-left: self;
margin-right:auto; padding: 10px;
border: 1px solid #000;}.form-content { padding: 5px;}input { width: 95%; padding: 5px; margin-
top: 10px; margin-bottom: 10px;}
label { font-size: 20px;}h1 { font-size: 24px; text-align: center;}.select { width: 100%; padding: 10px;
margin-top: 10px; margin-bottom: 10px; font-size: 18px;}.submit { width: 100%; font-size: 23px;
background: #000; color: #fff; cursor: pointer; padding: 5px;}.message { text-align: center; font-
size: 33px; margin-top: 5%; margin-bottom: 10px;}.error-link { text-align: center; font-size:
23px;}.error-link to { text-decoration: none;}
```

```
</style>
```

```
</head> <body>
```

```

<?php
if (isset($m)) {

if ($m==1) { echo '<div class="message">Payment Accepted Successfully!</div> <div class="error-link"><a href=".">Test Again</ a></div> '; } elseif ($m==2) { echo '<div class="message" id="error">There Was A Problem With Your CC</div> <div class="error-link"><a href=".">Click Here To Try Again</a></div> '; }

} else {?>

<div class="content-wrapper"> <h1>Billing Information</h1> <form action="payment.php" method="POST" autocomplete="off"> <div class="form-content">

<label>Amount</label> <input type="text" name="amount" value="98.61"/>

<label>Your Full Name</label> <input type="text" name="name" value="John Smith"/>

<label>Your Email Address</label> <input type="text" name="email" value="johns99@email.com"/>

<label>Street Address</label> <input type="text" name="street" value="1841 Wilbor Avenue"/>

<label>City</label> <input type="text" name="city" value="North Port"/>

<label>State</label> <select class="select" name="state"> <option value="AL">Alabama</option>
<option value="AK">Alaska</option> <option value="AZ">Arizona</option>
<option value="AR">Arkansas</option> <option value="CA">California</option> <option value="CO">Colorado</option> <option value="CT">Connecticut</ option> <option value="DE">Delaware</option> <option value="DC">District Of Columbia</option> <option value="FL">Florida</option> <option value="GA">Georgia</option> <option value="HI">Hawaii</option> <option value="ID">Idaho</option> <option value="IL">Illinois</option> <option value="IN">Indiana</option> <option value="IA">Iowa</option> <option value="KS">Kansas</option> <option value="KY">Kentucky</option> <option value="LA">Louisiana</option> <option value="ME">Maine</option> <option value="MD">Maryland</option> <option value="MA">Massachusetts</option> <option value="MI">Michigan</ option> <option value="MN">Minnesota</option> <option value="MS">Mississippi</option> <option value="MO">Missouri</option> <option value="MT">Montana</option> <option value="NE">Nebraska</option> <option value="NV">Nevada</option> <option value="NH">New Hampshire</option> <option value="NJ">New Jersey</option> <option value="NM">New Mexico</option> <option value="NY">New York</option> <option value="NC">North Carolina</option> <option value="ND">North Dakota</option> <option value="OH">Ohio</option> <option value="OK">Oklahoma</option> <option value="OR">Oregon</option> <option value="PA">Pennsylvania</option> <option value="RI">Rhode Island</option> <option value="SC">South Carolina</option> <option value="SD">South Dakota</option> <option value="TN">Tennessee</option> <option value="TX">

```



```

selected>Texas</option> <option value="UT">Utah</option> <option
value="VT">Vermont</option> <option value="VA">Virginia</option> <option
value="WA">Washington</option> <option value="WV">West Virginia</option>
<option value="WI">Wisconsin</option> <option value="WY">Wyoming</option> </select>

<label>Zip Code</label> <input type="text" name="zip" value="75001"/>

<h1>Credit Card Info</h1>

<label>Card Number</label> <input type="text" name="card" value="4111111111111111"/>

<label>Expiration Month</label> <select name="card_exp_month" class="select"> <option
value="01">01 Jan</option> <option value="02">02 Feb</option> <option value="03">03
Mar</option> <option value="04">04 Apr</option> <option value="05">05 May</option>
<option value="06">06 Jun</option> <option value="07">07 Jul</option> <option value="08">08
Aug</option> <option value="09">09 Sep</option> <option value="10">Oct 10</option> <option
value="11">Nov 11</option> <option value="12" selected>Dec 12</option> </select>

<label>Expiration Year</label> <select name="card_exp_year" class="select"> <?php echo
$firstYear = (int)date('Y'); $lastYear = $firstYear + 10; for($i=$firstYear;$i<=$lastYear;$i++) { echo
'<option value=".' . $i . '>.' . $i . '</option>'; } ?> </select>

<label>CVV</label> <input type="text" name="cvv" value="186"/>

<input type="submit" name="submit" class="submit" value="PAY NOW">

</div> </form></div><?php }?>

</body></html>

```

The important thing about this code is that the form in its submit action is directed to the file that we created with the help of the sandbox that in our case we call payments.

Now we proceed to launch the project with XAMPP and view the generated interface through the localhost.

Billing Information

Amount

99.61

Your Full Name

John Smith

Your Email Address

johns99@gmail.com

Street Address

1841 Wilbor Avenue

City

North Port

State

Texas

Zip Code

75001

Credit Card Info

Card Number

4111111111111111

Expiration Month

12 Dec

Expiration Year

2024

CVV

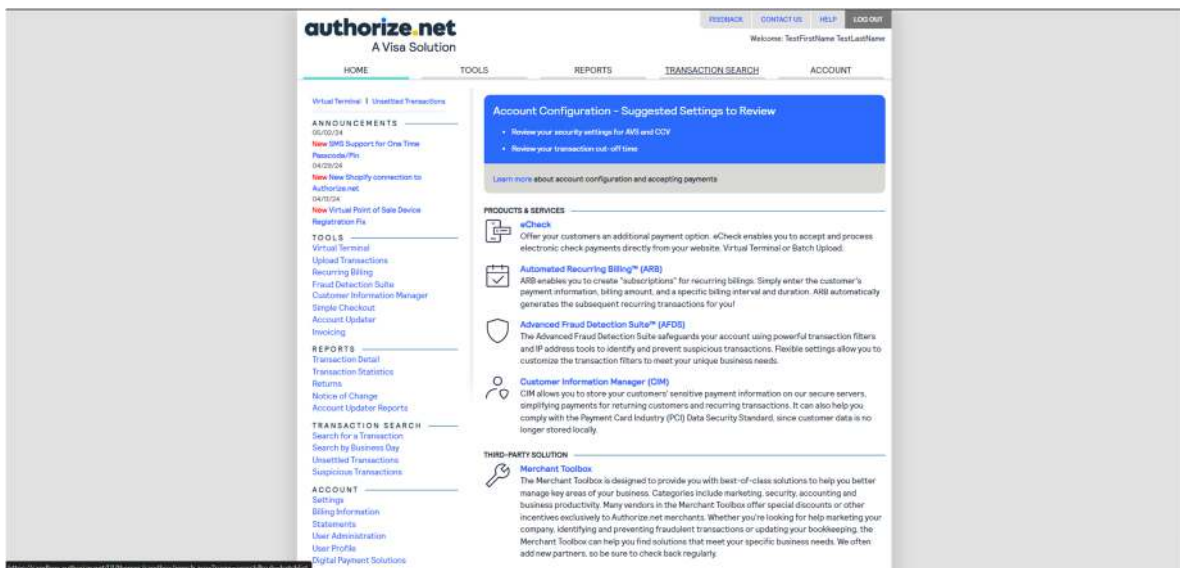
186

PAY NOW

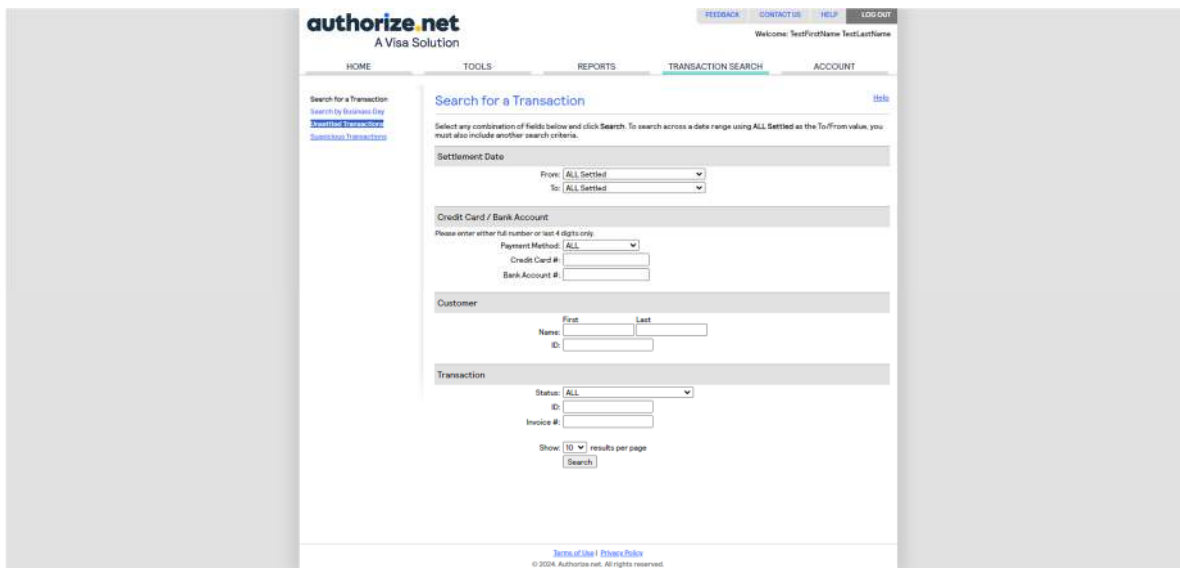
The data provided in the interface fields are for testing purposes and will be used to check through the sandbox that the transaction was successful or, if applicable, will display the process message.

Payment Accepted Successfully!
[Test Again](#)

To verify on the same Sandbox page we can view the processes that were requested.



1. Transaction Search Tab



2. Unsettled Transactions Option

authorize.net

A Visa Solution

TESTBACK

CONTACT US

HELP

LOG OUT

Welcome TestFirstName TestLastName

HOME

TOOLS

REPORTS

TRANSACTION SEARCH

ACCOUNT

Search + Unsettled Transactions

Unsettled Transactions

Select an action in the Filter by drop-down box to capture or void multiple transactions. Select the Trans ID from the list of transactions to view details, capture, or void a specific transaction.

Filter by ALL View

Use of Unsettled Transactions
Click on a transaction ID below to void transaction or view transaction details such as reason for decline. Click on any column heading to sort.

Note: If your account is in TEST MODE, transactions submitted for capture or void will NOT actually be processed. Before submitting live transactions, verify that TEST MODE is turned off.

Filter by ALL View Print

1-2 of 2 results

Trans ID	Invoice Number	Trans Status	Settled Date	Customer	Card	Payment Method	Payment Amount
8802000552	686712	Captured/Pending Settlement	20-Sep-2024 21:55:64	John Smith	V	XXXX1111	USD 45.60
8802000553	686712	Captured/Pending Settlement	20-Sep-2024 21:57:35	John Smith	V	XXXX1111	USD 98.61

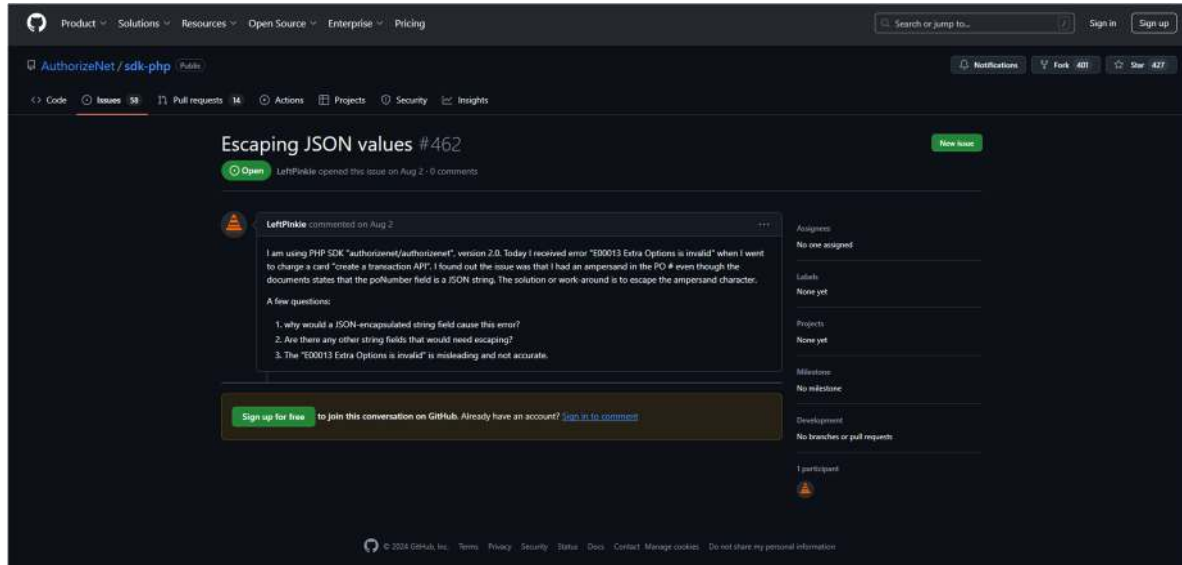
1-2 of 2 results

Search + Unsettled Transactions

Terms of Use | Privacy Policy

© 2024, Authorize.net. All rights reserved.

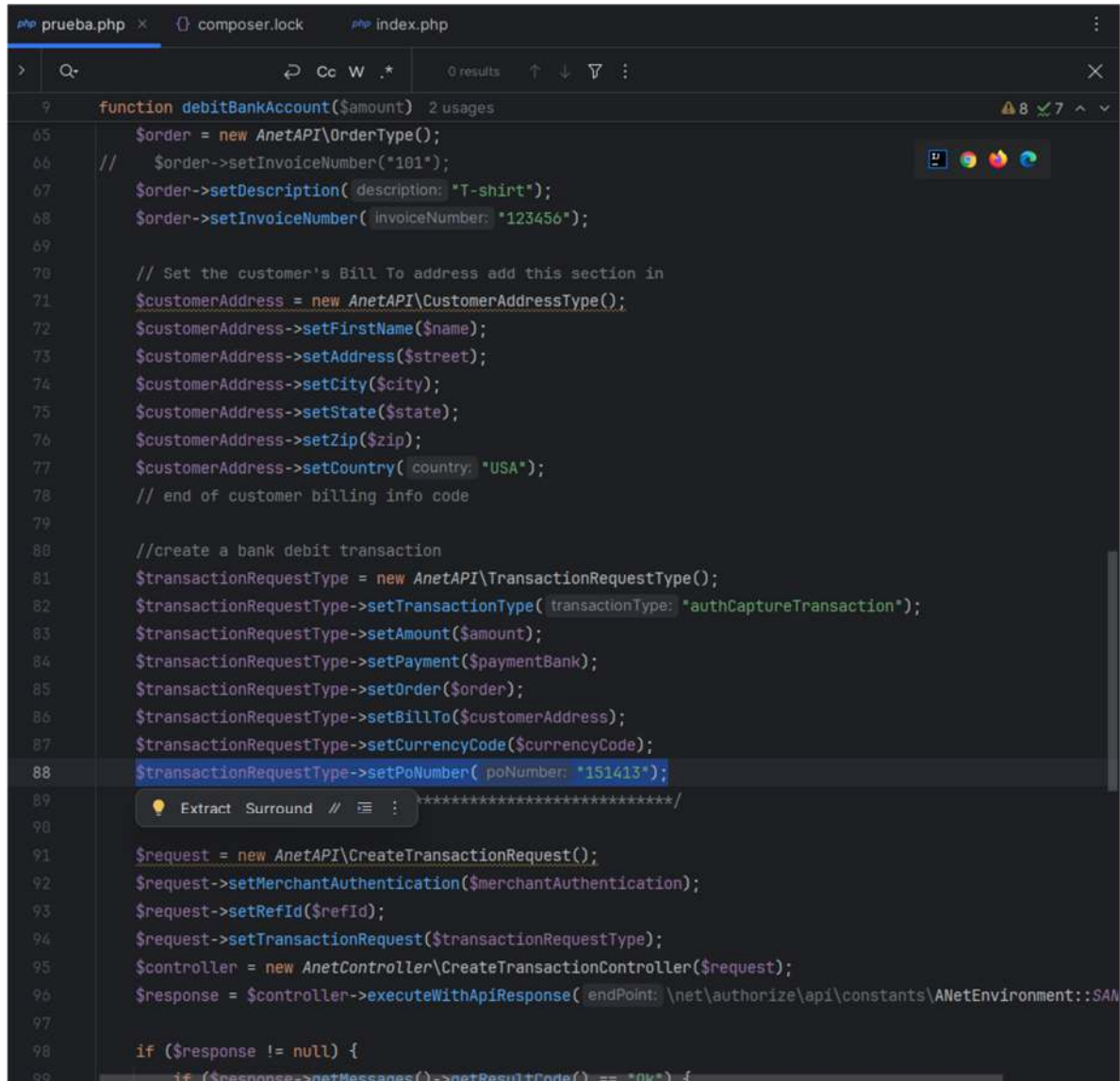
Issue 1 - 462 -Escaping JSON values



Cause and solution:

This issue mentions that when sending the poNumber, an attempt was made to include the # symbol and this caused an error when trying to carry out the transaction. The 2 tests were carried out to verify if this error continued and in neither case did it cause errors, so we can conclude that the problem could have come from another bad configuration by the developer or it was previously solved by the SDK.

Evidence



The screenshot shows a code editor with a dark theme. The top bar indicates the file is 'prueba.php' and shows a search bar with '0 results'. The code is a PHP function named 'debitBankAccount' that takes '\$amount' as a parameter. It creates an 'AnetAPI\OrderType' object, sets its description to 'T-shirt' and invoice number to '123456'. It then creates a 'CustomerAddressType' object and sets its details (name, address, city, state, zip, country). Next, it creates a 'TransactionRequestType' object, sets its transaction type to 'authCaptureTransaction', amount to '\$amount', payment to '\$paymentBank', order to '\$order', bill to '\$customerAddress', and currency code to '\$currencyCode'. Line 88, which is highlighted, sets the 'poNumber' to '151413'. The code then creates a 'CreateTransactionRequest' object, sets its merchant authentication, reference ID, and transaction request. It creates a 'CreateTransactionController' and executes it with the request. Finally, it checks if the response is not null and if the result code is '0x'.

```
9 function debitBankAccount($amount) 2 usages
65 $order = new AnetAPI\OrderType();
66 // $order->setInvoiceNumber("101");
67 $order->setDescription( description: "T-shirt");
68 $order->setInvoiceNumber( invoiceNumber: "123456");
69
70 // Set the customer's Bill To address add this section in
71 $customerAddress = new AnetAPI\CustomerAddressType();
72 $customerAddress->setFirstName($name);
73 $customerAddress->setAddress($street);
74 $customerAddress->setCity($city);
75 $customerAddress->setState($state);
76 $customerAddress->setZip($zip);
77 $customerAddress->setCountry( country: "USA");
78 // end of customer billing info code
79
80 //create a bank debit transaction
81 $transactionRequestType = new AnetAPI\TransactionRequestType();
82 $transactionRequestType->setTransactionType( transactionType: "authCaptureTransaction");
83 $transactionRequestType->setAmount($amount);
84 $transactionRequestType->setPayment($paymentBank);
85 $transactionRequestType->setOrder($order);
86 $transactionRequestType->setBillTo($customerAddress);
87 $transactionRequestType->setCurrencyCode($currencyCode);
88 $transactionRequestType->setPoNumber( poNumber: "151413");
89
90
91 $request = new AnetAPI\CreateTransactionRequest();
92 $request->setMerchantAuthentication($merchantAuthentication);
93 $request->setRefId($refId);
94 $request->setTransactionRequest($transactionRequestType);
95 $controller = new AnetController\CreateTransactionController($request);
96 $response = $controller->executeWithApiResponse( endPoint: \net\authorize\api\constants\ANetEnvironment::SAND
97
98 if ($response != null) {
99     if ($response->getMessages()->getResultCode() == "0x") {
```

Pay Invoice



Payment amount

\$ 30

Name Bank

Santander

Name Account

Luis Fernando Vargas

Card number

6666666666666666

State

Camp.

City

Carmen

Country USA ▼

USD ▼

Address

Tijerilla

ZIP/Postal code

24180



Pay

Transaction Response code : 1 Debit Bank Account APPROVED : Debit Bank Account AUTH CODE : Debit Bank Account TRANS ID : 80026045071 Code : 1 Description : This transaction has been approved.

authoriza
A Visa

HOME

Search > Unsettled Transactions

Unsettled Transactions

Select an action in the Filter
a specific transaction

Filter by

List of Unsettled Transactions
Click on a transaction ID below

Note: If your account is in TEST
MODE is turned off

Filter by:

1-3 of 3 results

Transaction ID	Invoice Number
80026045071	123456
80026045072	123456
10040024348	101

1-3 of 3 results

Search > Unsettled Transactions

Transaction Detail

Transaction ID: 80026045071

Transaction Status: Captured/Pending Settlement

VoidPrintClose

Settlement Information

Settlement Amount: USD 30.00

Settlement Date and Time:

Authorization Information

Authorization Amount: USD 30.00

Submit Date/Time: 24-Sep-2024 16:40:36 PDT

Reference Transaction ID: Not Applicable

Transaction Type: Authorization w/ Auto Capture

Recurring Billing Transaction: N

Customer IP: 188.180.250.223

Payment Information

Card Type: eCheck

Account Number: XXXX6666

ABA Routing Number: XXXX0688

Name on Bank Account: Luis Fernando Vergas

eCheck Type: WEB

Total Amount: USD 30.00

Order Information

Invoice #: 123456

Description: T-shirt

Customer Billing Information

Name: Luis Fernando vergas

Company:

Address: Tijerilla

City: Carmen

State/Province: Camp

Zip Code: 24180

Country: USA

Phone:

Fax:

Email:

Customer ID:

Customer Type:

Customer TaxID/SSN:

DL Number:

DL State:

DL DOB:

Shipping Information

Name:

Company:

Address:

City:

State/Province:

Zip Code:

Country:

Additional Details - Level 2 Data

Tax: 0.00

Freight: 0.00

Duty: 0.00

Tax Exempt: N

PO Number: 151413

VoidPrintClose

ACCOUNT

TestFirstName TestLastName

view details, capture, or void

ort.

actions, verify that TEST

Print

Method	Payment Amount
XXX6666	USD 30.00
XXX5555	USD 30.00
XXX4444	USD 30.00

Pay Invoice



Payment amount

\$ 30

Name Bank

Santander

Name Account

Luis Fernando Vargas

Card number

1111111111111111

State

Camp.

City

Carmen

Country USA ▼

USD ▼

Address

Tijerilla

ZIP/Postal code

24180



Pay

Transaction Response code : 1 Debit Bank Account APPROVED : Debit Bank Account AUTH CODE : Debit Bank Account TRANS ID : 80026045222 Code : 1 Description : This transaction has been approved.

```
> Q- Cc W .* 0 results ↑ ↓ V :
9 function debitBankAccount($amount) 2 usages
65 $order = new AnetAPI\OrderType();
66 // $order->setInvoiceNumber("101");
67 $order->setDescription( description: "T-shirt");
68 $order->setInvoiceNumber( invoiceNumber: "123456");
69
70 // Set the customer's Bill To address add this section in
71 $customerAddress = new AnetAPI\CustomerAddressType();
72 $customerAddress->setFirstName($name);
73 $customerAddress->setAddress($street);
74 $customerAddress->setCity($city);
75 $customerAddress->setState($state);
76 $customerAddress->setZip($zip);
77 $customerAddress->setCountry( country: "USA");
78 // end of customer billing info code
79
80 //create a bank debit transaction
81 $transactionRequestType = new AnetAPI\TransactionRequestType();
82 $transactionRequestType->setTransactionType( transactionType: "authCaptureTransaction");
83 $transactionRequestType->setAmount($amount);
84 $transactionRequestType->setPayment($paymentBank);
85 $transactionRequestType->setOrder($order);
86 $transactionRequestType->setBillTo($customerAddress);
87 $transactionRequestType->setCurrencyCode($currencyCode);
88 $transactionRequestType->setPoNumber( poNumber: "#151413");
89 Extract Surround // ⋮ *****/
90
91 $request = new AnetAPI\CreateTransactionRequest();
92 $request->setMerchantAuthentication($merchantAuthentication);
93 $request->setRefId($refId);
94 $request->setTransactionRequest($transactionRequestType);
95 $controller = new AnetController\CreateTransactionController($request);
96 $response = $controller->executeWithApiResponse( endPoint: \net\authorize\api\constants\ANetEnvironment:
97
98 if ($response != null) {
```

authoriz

A Visa

[Feedback](#)
[Contact Us](#)
[Help](#)
[Logout](#)

Transaction Detail

Transaction ID: 80026040222

[Create ABB Subscription from Transaction](#)

Transaction Status: Captured/Pending Settlement

Void

Print

Close

Settlement Information

Settlement Amount: USD 30.00

Settlement Date and Time:

Authorization Information

Authorization Amount: USD 30.00

Subm Date/Time: 24-Sep-2024 16:50:36 PDT

Reference Transaction ID: Not Applicable

Transaction Type: Authorization w/ Auto Capture

Recurring Billing Transaction: N

Customer ID: 186380250223

Payment Information

Card Type: eCheck

Account Number: XXXX0111

ABA Routing Number: XXXX0681

Name on Bank Account: Luis Fernando Vargas

eCheck Type: WEB

Total Amount: USD 30.00

Order Information

Invoice #: 123456

Description: T-shirt

Customer Billing Information

Name: Luis Fernando Vargas

Company:

Address: Timilla

City: Ciemien

State/Province: Camp

Zip Code: 24180

Country: USA

Phone:

Fax:

Email:

Customer ID:

Customer Type:

Customer Tax/USIN:

DL Number:

DL State:

DL DOB:

Shipping Information

Name:

Company:

Address:

City:

State/Province:

Zip Code:

Country:

Additional Details - Level 2 Data

Tax: 0.00

Freight: 0.00

Duty: 0.00

Tax Exempt: N

PO Number: #101413

Void

Print

Close

HOME

Search > Unsettled Transactions

Unsettled Transactions

Select an action in the Filter specific transaction.

Filter by:

List of Unsettled Transactions

Note: If your account is in TEST MODE is buried off.

Filter by:

Search ID

Transaction ID

Transaction Amount

Transaction Date

Transaction Status

80026040222

123456

USD 30.00

24-Sep-2024

Captured

80026040223

123456

USD 30.00

24-Sep-2024

Captured

80026040224

123456

USD 30.00

24-Sep-2024

Captured

80026040225

123456

USD 30.00

24-Sep-2024

Captured

80026040226

123456

USD 30.00

24-Sep-2024

Captured

80026040227

123456

USD 30.00

24-Sep-2024

Captured

80026040228

123456

USD 30.00

24-Sep-2024

Captured

80026040229

123456

USD 30.00

24-Sep-2024

Captured

80026040230

123456

USD 30.00

24-Sep-2024

Captured

ACCOUNT

TestFirstName TestLastName

view details, capture, or void

transactions, verify that TEST

Print

Search ID

Transaction ID

Transaction Amount

Transaction Date

Transaction Status

80026040231

123456

USD 30.00

24-Sep-2024

Captured

80026040232

123456

USD 30.00

24-Sep-2024

Captured

80026040233

123456

USD 30.00

24-Sep-2024

Captured

80026040234

123456

USD 30.00

24-Sep-2024

Captured

80026040235

123456

USD 30.00

24-Sep-2024

Captured

80026040236

123456

USD 30.00

24-Sep-2024

Captured

80026040237

123456

USD 30.00

24-Sep-2024

Captured

80026040238

123456

USD 30.00

24-Sep-2024

Captured

80026040239

123456

USD 30.00

24-Sep-2024

Captured

80026040240

123456

USD 30.00

24-Sep-2024

Captured

Issue 5 - 446 -Payment but decline receive payment in my authorize wallet

Cause and solution:

In this case the problem to start with is an error in the developer's code in which he tries to display a message through the logical function with the following code:

```
if (($response != null) && ($response->getResponseCode() == "1")) {
```

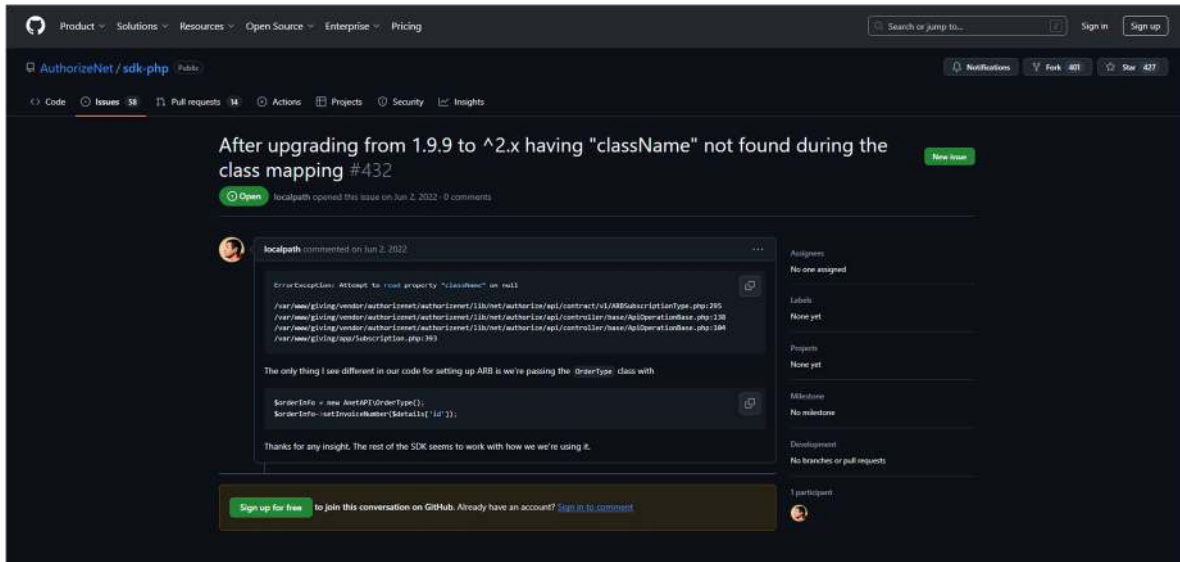
Which, although in logic it should not show an error, this way of invoking the result could be incorrect according to the API reference, which in its correct statement should be:

```
if ($response->getMessages()->getResultCode() == "Ok") {
```

As one user commented, you should check what message or process the Sandbox page displays to get a more precise idea of what could have caused the error reported.

We can obtain the evidence from the previous problem-solving cases.

Issue 10 - 432 -After upgrading from 1.9.9 to ^2.x having "className" not found during the class mapping



```

// $bankAccount->setAccountNumber(rand(10000,999999999999));
$bankAccount->setAccountNumber($accountNumber);

$bankAccount->setNameOnAccount($name);
$bankAccount->setBankName($nameAccount);

// $bankAccount->setNameOnAccount("John Doe");
// $bankAccount->setBankName('Wells Fargo Bank NA');

$paymentBank= new AnetAPI\PaymentType();
$paymentBank->setBankAccount($bankAccount);

// Order info
$order = new AnetAPI\OrderType();
// $order->setInvoiceNumber("101");
$order->setDescription( description: "T-shirt");
$order->setInvoiceNumber( invoiceNumber: "123456");

// Set the customer's Bill To address add this section in
$customerAddress = new AnetAPI\CustomerAddressType();
$customerAddress->setFirstName($name);
$customerAddress->setAddress($street);
$customerAddress->setCity($city);
$customerAddress->setState($state);
$customerAddress->setZip($zip);
$customerAddress->setCountry( country: "USA");
// end of customer billing info code

//create a bank debit transaction
$transactionRequestType = new AnetAPI\TransactionRequestType();
$transactionRequestType->setTransactionType( transactionType: "authCaptureTransaction");
$transactionRequestType->setAmount($amount);
$transactionRequestType->setPayment($paymentBank);
$transactionRequestType->setOrder($order);
$transactionRequestType->setBillTo($customerAddress);

```

Pay Invoice

VISA MASTERCARD AMERICAN EXPRESS

Payment amount
\$ 30

Name Bank
Santander

Name Account
Luis Fernando Vargas

Card number
5555555555555555

State
Queretaro

City
Queretaro

Country USA USD

Address
Capuchinos

ZIP/Postal code
7903

Pay

Transaction Response code : 1 Debit Bank Account APPROVED : Debit Bank Account AUTH CODE : Debit Bank Account TRANS ID : 80026043572 Code : 1 Description : This transaction has been approved.

Solution:

In `phplog.php` we get proof of sending

Wed, 25 Sep 2024 00:21:26 +0200 INFO : [_sendRequest] (C:\xampp\htdocs\PHP_USAA\sdk-php-master\lib\net\authorize\util\HttpClient.php : 77) - Request to AnetApi:

```
{
  "createTransactionRequest": {
    "merchantAuthentication": {
      "name": "4U293qwGz",
      "transactionKey": "433dcue5992EP4FC"
    },
    "clientId": "sdk-php-2.0.2",
    "refId": "ref1727216486",
    "transactionRequest": {
      "transactionType": "authCaptureTransaction",
      "amount": "30",
      "currencyCode": "USD",
      "payment": {
        "bankAccount": {
          "accountType": "checking",
          "routingNumber": "122000661",
          "accountNumber": "xxxx",
          "nameOnAccount": "Luis Fernando Vargas",
          "echeckType": "WEB",
          "bankName": "Santander"
        },
        "order": {
          "invoiceNumber": "123456"
        },
        "description": "T-shirt",
        "billTo": {
          "firstName": "Luis Fernando Vargas",
          "address": "Capuchinos",
          "city": "Queretaro",
          "state": "Queretaro",
          "zip": "7903",
          "country": "USA"
        }
      }
    }
  }
}
```


Evidence

Transaction ID: 80026043572 [Create ARB Subscription from Transaction](#)
Transaction Status: Captured/Pending Settlement

Settlement Information

Settlement Amount: USD 30.00
Settlement Date and Time:

Authorization Information

Authorization Amount: USD 30.00
Submit Date/Time: 24-Sep-2024 15:21:28 PDT
Reference Transaction ID: Not Applicable
Transaction Type: Authorization w/ Auto Capture
Recurring Billing Transaction: N
Customer IP: 189.180.250.223

Payment Information

Card Type: eCheck
Account Number: XXXX5555
ABA Routing Number: XXXX0661
Name on Bank Account: Luis Fernando Vargas
eCheck Type: WEB
Total Amount: USD 30.00

Order Information

[Invoice #](#): 123456
Description: T-shirt

Customer Billing Information

Name: Luis Fernando Vargas
Company:
Address: Capuchinos
City: Queretaro
State/Province: Queretaro
Zip Code: 7903
Country: USA
Phone:
Fax:
Email:

Customer ID:
Customer Type:
Customer TaxID/SSN:

DL Number:
DL State:
DL DOB:

Shipping Information

As you can see in the screenshots the invoiceNumber value is processed without any problem, and the error mentioned in the issue is likely caused by an error in the data or its parameters according to AuthorizeNet documentation.

• • invoiceNumber Merchant-defined invoice number associated with the order. String, up to 20 characters.

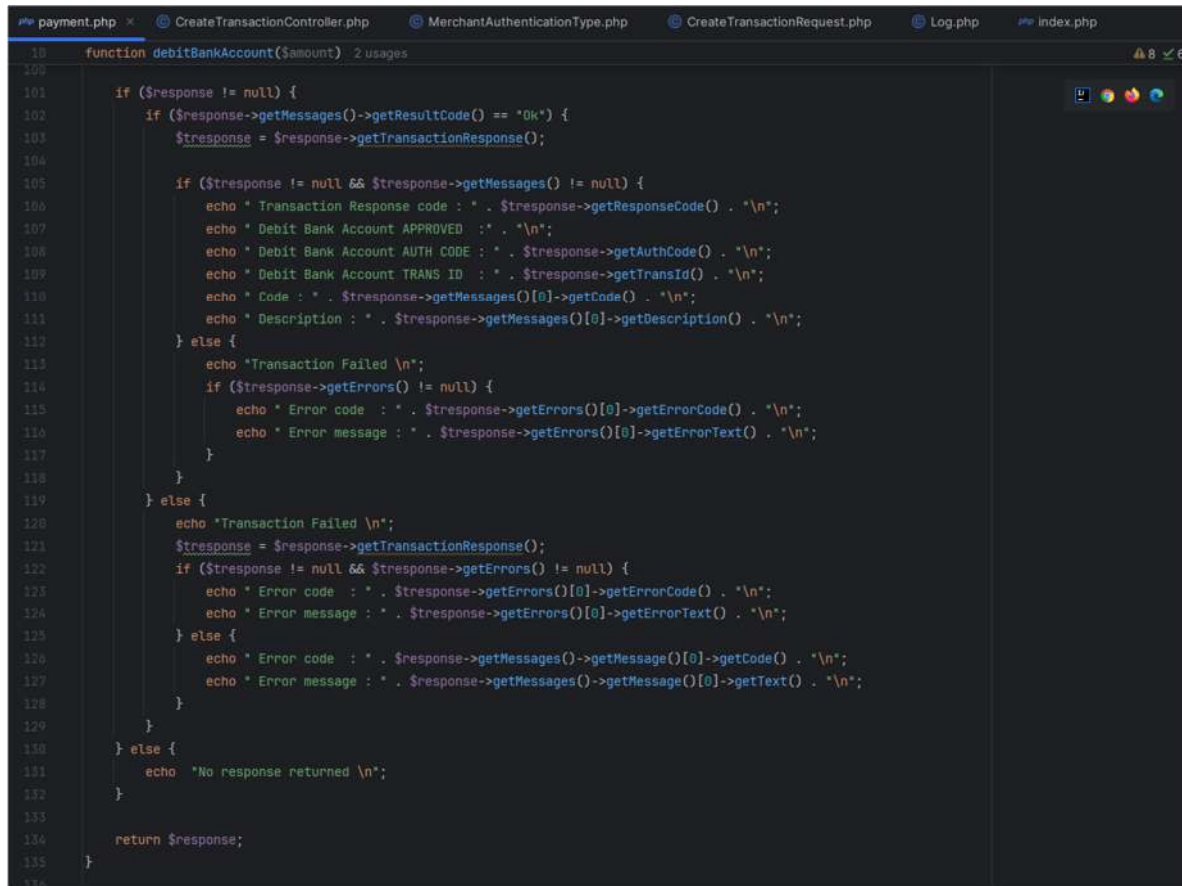
Worldpay RAFT 610 merchants can view the invoice number in the Worldpay Reporting Portal.

Issue 11 – 430 – CreateTransitionController returns null, conflicting examples

Cause and Solution:

In this issue the user mentions that he gets a null value in response from the Sandbox API, but from what other users are saying the problem seems to be a server crash on the part of SandBox since they say that it only happened once.

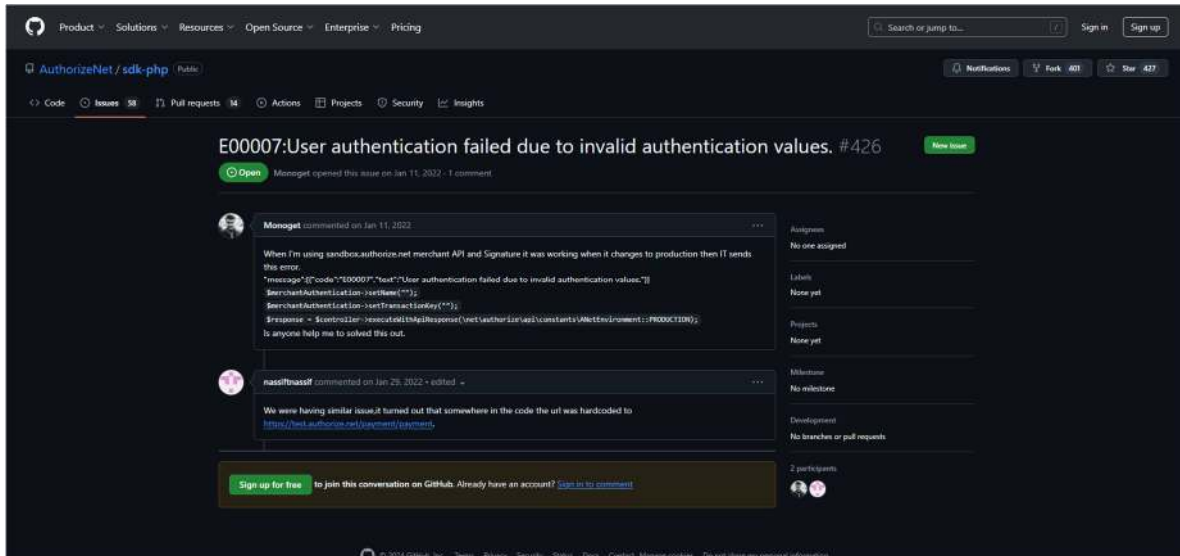
I am attaching the same code as the one mentioned by the first user where you can verify that the SDK works correctly.



```
18 function debitBankAccount($amount) 2 usages
199
101 if ($response != null) {
102     if ($response->getMessages()->getResultCode() == "OK") {
103         $response = $response->getTransactionResponse();
104
105         if ($response != null && $response->getMessages() != null) {
106             echo "Transaction Response code : " . $response->getResponseCode() . "\n";
107             echo "Debit Bank Account APPROVED : " . "\n";
108             echo "Debit Bank Account AUTH CODE : " . $response->getAuthCode() . "\n";
109             echo "Debit Bank Account TRANS ID : " . $response->getTransId() . "\n";
110             echo "Code : " . $response->getMessages()[0]->getCode() . "\n";
111             echo "Description : " . $response->getMessages()[0]->getDescription() . "\n";
112         } else {
113             echo "Transaction Failed \n";
114             if ($response->getErrors() != null) {
115                 echo "Error code : " . $response->getErrors()[0]->getErrorCode() . "\n";
116                 echo "Error message : " . $response->getErrors()[0]->getErrorMessage() . "\n";
117             }
118         }
119     } else {
120         echo "Transaction Failed \n";
121         $response = $response->getTransactionResponse();
122         if ($response != null && $response->getErrors() != null) {
123             echo "Error code : " . $response->getErrors()[0]->getErrorCode() . "\n";
124             echo "Error message : " . $response->getErrors()[0]->getErrorMessage() . "\n";
125         } else {
126             echo "Error code : " . $response->getMessages()->getMessage()[0]->getCode() . "\n";
127             echo "Error message : " . $response->getMessages()->getMessage()[0]->getText() . "\n";
128         }
129     }
130 } else {
131     echo "No response returned \n";
132 }
133
134 return $response;
135 }
```

This code is from the file called payments that was obtained from the API as mentioned in the previous reports and contains the configuration for the transaction to be carried out.

Issues 13 - 426 - E00007: User authentication failed due to invalid authentication values



Cause and Solution:

This issue mentions that it marks an error because the connection was changed to PRODUCTION. This is because in the user tests the SANDBOX connection is used which allows transactions without charging any card, so it is most likely due to incorrect data in the NAME and TRANSACTIONKEY parameters, data necessary for the connection with the AuthorizeNet API. This is not a code error.

<https://www.authorize.net/sign-up/pricing.html>

A search was carried out for users with the same problem and a similar issue was found on the STACKOVERFLOW page that has already been closed and possibly solved, mentioning the above.

stackoverflow

AboutProductsOverflowWiki

Search...

Log inSign up

Home

Questions

Tags

Users

Companies

LABS

Jobs

Discussions

RECENT TAGS

authorize.netphpregistr

COLLECTIVES

Communities for your favorite technologies: [Elasticsearch](#) [Elasticsearch](#)

TEAMS

overflow AI

Now available on Stack Overflow for Teams. A feature where you work, learn, and grow.

Learn more

Explore Teams

Authorize.Net payments not working in production mode

Asked 2 years agoModified 1 year, 11 months agoViewed 803 times

1 implemented a capture credit card with authnet and it worked fine in sandbox environment but now I have changed it to live.

0

\$response = \$controller->executeWithApiResponse(\$net\authorizeapi\constants\AllEnv

I have also changed my `$searchant` authentication values to production credentials:

SearchantAuthentication->setKey(Settings::get('auth_net_apl_id'));

SearchantAuthentication->setTransactionKey(Settings::get('auth_net_apl_key'));

When using test card I get the error:

An error occurred during processing. Call Merchant Service Provider.

When using actual card the request does not get handled and results in 500 error:

authorize.net

ShareImprove this questionFollow

edited Oct 7, 2022 at 14:53

John Conde

219k ● 99 ● 461 ● 501

asked Sep 26, 2022 at 7:14

Sakshi Gnananya

181 ● 7

ADD A COMMENT

1 Answer

Sorted by Highest score (default)

¿No encuentras la respuesta? Pregunte en Stack Overflow en español

According to [Authorize.Net's own documentation](#), the merchant account is set up incorrectly. You need to contact the merchant account provider and have them make whatever correction is needed for your integration to work.

2

ShareImprove this answerFollow

answered Sep 26, 2022 at 12:14

John Conde

219k ● 99 ● 461 ● 501

ADD A COMMENT

Your Answer

Sign up or log in

Post as a guest

Name

Email

Required, but never shown

Sign up using Google

Sign up using Email and Password

Post Your Answer

By clicking "Post Your Answer" you agree to our terms of service and acknowledge you have read our privacy policy.

Not the answer you're looking for? Browse other questions tagged [authorize.net](#) or ask your own question.

The Overflow Blog

Masked self-attention: How LLMs learn relationships between tokens

Deedy Day: from coding at Meta, to search at Google, to leavibg with Anthropic

Featured on Meta

User avatars: learnings and opportunities

Preventing unauthorized automated access to the network

Should low-scoring meta questions no longer be hidden on the MetaSO home...

Feedback Requested: How do you use the tagged questions page?

Announcing the new Staging Ground Reviewer Stats Widget

Linked

error code 25 - An error occurred during processing

Related

Authorize.net issues with American Express

Why does my Authorize.net test account fail using the sample app they provide?

authorize.net Production account accepts test credit card numbers

Authorize.net payment gateway issue

Authorize.net Payment in Magento does not show credit card form when selected

Active Merchant and Authorize.net returning failure in production

ARB failed payment

Authorize.net sandbox transactions API not working in live server

Authorize.net - how to make a payment on Test Mode

Payment with authorize.net gives error User authentication failed due to invalid authentication values

Hot Network Questions

Why would an e-commerce company not accept ACH payments?

Can we solve the 3-sat problem with probability?

Script on "Create public links for approved assets automatically" documentation page does not compile

How to sub-align expressions and preserve equation numbering?

Print 4 Billion R statements

more hot questions

STACK OVERFLOW

QuestionsHelpChat

PRODUCTS

IntegrationsAdvertisingJobs

COMPANY

AboutPressWork with usLegalPrivacy PolicyTerms of ServiceContact UsCreate SettingsFeedback

STACK EXCHANGE NETWORK

Technology

Culture & Recreation

Life & Arts

Science

Professional

Business

API

Data

Fast

Fun

Free

Helpful

Insightful

Stack Overflow

Stack Exchange

Stack Overflow

Stack Overflow

Stack Overflow

Stack Exchange

Stack Overflow

Stack Overflow

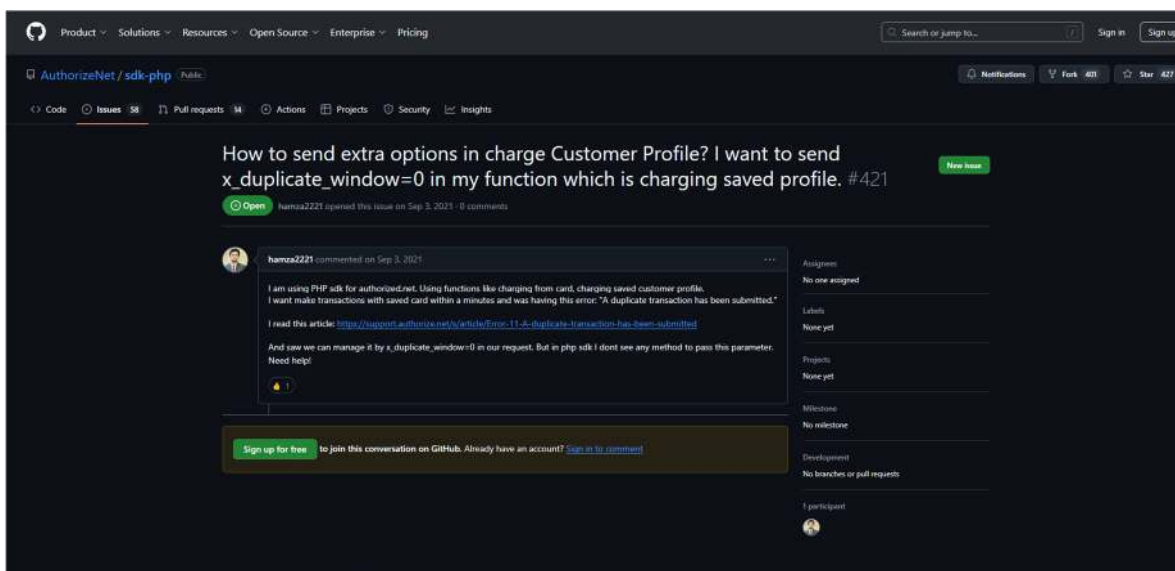
<https://stackoverflow.com/questions/73782870/authorize-net-payments-not-working-in-production-mode>

36

Issue 17 – 421 – how to send extra options in charge Customer Profile? I want to send x_duplicate_window=0 in my function which is charging saved profile

Cause:

This issue indicates that the user was successfully transacting, but started seeing duplicate transactions, possibly due to trying to test with the same data. Authorize.net identifies duplicate transactions by matching the data provided to the transaction.



Solution:

The solution as mentioned on the AuthorizeNet Support page is that by adding the following line of code: `x_duplicate_window` field to the website script, we can control a response from the server which tells us how long we should wait to check for duplicates after a transaction is sent. The support response with all the information is attached.

Error 11: Se ha enviado una transacción duplicada.

Número de artículo	Número total de visualizaciones del artículo	Artículos recomendados para ti
000001473	10145	¿Cómo puedo contactar con Authorize.net?
Fecha de última modificación	Versión	¿Cuál es el propósito del ID de inicio de sesión de API, la clave de transacción, la clave de firma y la clave pública para Authorize.net y cómo puedo obtenerlas?
18/07/2024 16:27	7.0	Migration de complementos de Shopify

Código de motivo de respuesta: 11

Texto del motivo de la respuesta: Se ha enviado una transacción duplicada.

Notas: Se envió una transacción con monto idéntico e información de tarjeta de crédito dentro de la ventana de Transacción duplicada.

El error indica que se envió una solicitud de transacción unos minutos después de un intento anterior con la misma información. Authorize.net identifica transacciones duplicadas haciendo coincidir los datos proporcionados con la transacción.

Los campos de datos que se validan son:

- ID de inicio de sesión de API (*i_login*)
- Número de tarjeta de crédito (*i_card_num*)
- Fecha de vencimiento (*i_exp_date*)
- Tipo de transacción (*i_type*)
- Número de cuenta bancaria (*i_bank_acct_num*)
- Número de ruta (*i_bank_atm_code*)
- Número de orden de compra (*i_ord_num*)
- Cantidad (*i_qty*)
- Número de factura (*i_invoice_num*)
- Identificación del cliente (*i_cust_id*)
- Nombre (*i_first_name*)
- Apellido (*i_last_name*)
- Compañía (*i_company*)
- Dirección (*i_address*)
- Ciudad (*i_city*)
- Estado (*i_state*)
- Código postal (*i_zip*)
- País (*i_country*)
- Ventana duplicada (*i_duplicate_window*)

Si alguno de los campos cambia de una transacción a la siguiente, Authorize.net no verá las transacciones como duplicadas.

La ventana de transacción duplicada siempre será de dos minutos para todas las transacciones enviadas a través de la Terminal Virtual. Si desea ajustar la ventana de transacción duplicada para las transacciones enviadas desde su software, como un sitio web o un carrito de compras, puede hacerlo agregando el campo `x_duplicate_window` al script de su sitio web.

Si no puede ajustar o agregar esta variable a la configuración de su carrito de compras, comuníquese con el equipo de soporte de su carrito de compras para obtener ayuda adicional al respecto. La variable `x_duplicate_window` nos indica, en segundos, ¿cuánto tiempo debemos esperar para verificar si hay duplicados después de que se envía una transacción?


El valor máximo que aceptaremos para `x_duplicate_window` es 28800, lo que equivale a ocho horas. Si se envía un valor mayor que 28800, la pasarela de pago utilizará 28800 como valor predeterminado. Si `x_duplicate_window` se configura en 0 o en un número negativo, no se aplicará ninguna ventana de transacción duplicada para las transacciones de su software. Si no se envía ningún valor, se utilizará el valor predeterminado de 120 (dos minutos).

Por ejemplo, si desea establecer una ventana de transacción duplicada de cinco minutos, deberá establecer `x_duplicate_window` en 300, ya que hay 300 segundos en cinco minutos.

Nota: Al enviar `x_duplicate_window` con sus transacciones en línea, le devolveremos más detalles junto con esta respuesta de error, que incluye:

- El ID de transacción original que se rechazó.
- Requisitos AVS y CVV para la transacción original.
- El código de autorización original, si la transacción fue autorizada.
- El hash MD5, si se generó un valor hash MD5 para la transacción original.

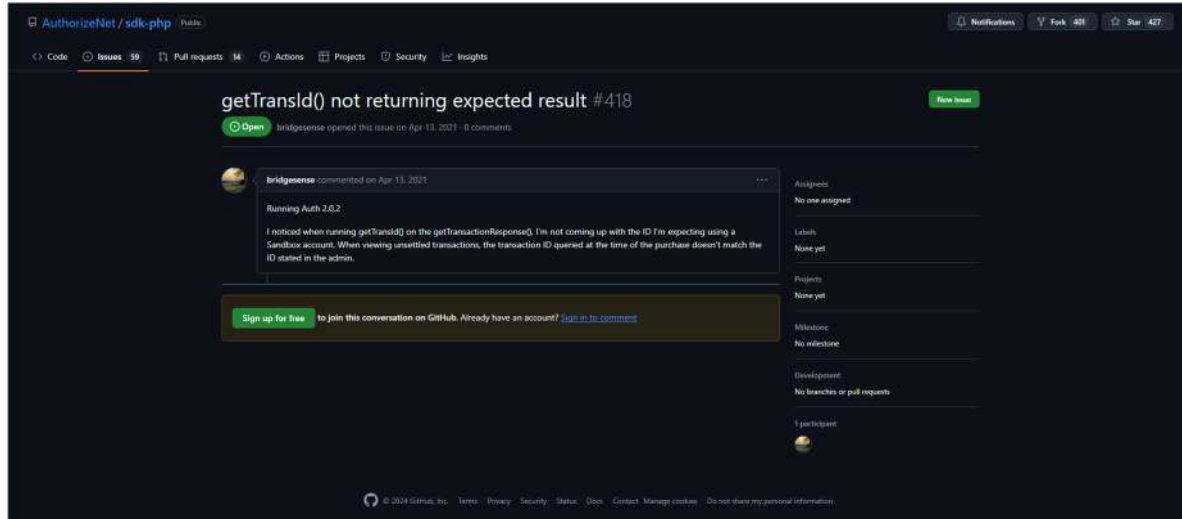
Si no envía el campo `x_duplicate_window`, no devolveremos ningún detalle de la transacción original, incluso si envía una transacción duplicada.

¿Fue útil este artículo?  

Issue 18 – 418 –getTransId() not returning expected result

Cause and solution:

In the issue it is stated that the value of getTransactionID is not obtained, but that in reality in the code used and with the examples it is received and shown in the response, and it has also been verified with the SANDBOX payment system that they match.



Transaction Response code : 1 Debit Bank Account APPROVED : Debit Bank Account AUTH CODE : Debit Bank Account **TRANS ID : 80026397894** Code : 1 Description : This transaction has been approved.

Transaction Detail

[Help](#)

Transaction ID: 80026397894

[Create ARB Subscription from Transaction](#)

Transaction Status: Captured/Pending Settlement

Void

Print

Close

Settlement Information

Settlement Amount: USD 30.00

Settlement Date and Time:

Authorization Information

Authorization Amount: USD 30.00

Submit Date/Time: 30-Sep-2024 06:04:00 PDT

Reference Transaction ID: Not Applicable

Transaction Type: Authorization w/ Auto Capture

Recurring Billing Transaction: N

Customer IP: 189.180.231.165

Payment Information

Card Type: eCheck

Account Number: XXXX2222

ABA Routing Number: XXXX0661

Name on Bank Account: Luis Fernando Vargas

eCheck Type: WEB

Total Amount: USD 30.00

Order Information

Invoice #: 123456

Description: T-shirt

Customer Billing Information

Name: Luis Fernando Vargas

Company:

Address: Capuchinos 83

City: MISION DE SAN CARLOS-CORREGIDORA

State/Province: Gro.

Zip Code: 76903

Country: USA

Phone:

Fax:

Email:

Customer ID:

Customer Type:

Customer TaxID/SSN:

Issue 24 – 411 - Other payment types are not supported?

AutotizeNet Payment Methods

The AuthorizeNet SDK allows for several types of payments:

Accept these payment types wherever you do business

Accept the payment types customers prefer. We support multiple forms of online payment*, so your customers can choose the payment types they prefer. Payment processing is simplified, so the customer experience is seamless whether they choose to pay online or in person.

VISA

mastercard

DISCOVER

AMERICAN EXPRESS

echeck

JCB

PayPal

Apple Pay

<https://www.authorize.net/resources/our-features/payment-types.html>

In our case we will be configuring the two essential methods that are VISA and eCheck and their operating parameters.

Payment Method – Debit (eCheck)

On the developer page we can find the correct way to configure the php file that we must integrate into the SDK, its requirements and appropriate parameters so that SandBox can process the payment.

<https://developer.authorize.net/api/reference/index.html#payment-transactions-debit-a-bank-account>

In the left side menu section, go to Debit a Bank Account, where you can find all the necessary data.

The screenshot shows the Anet API documentation for the 'Debit a Bank Account' endpoint. On the left is a sidebar with a menu under 'EXPAND ALL' containing categories like 'API Endpoints & Authentication', 'Payment Transactions', 'Mobile In-App Transactions', 'PayPal Express Checkout', 'Fraud Management', 'Recurring Billing', 'Customer Profiles', and 'Transaction Reporting'. The 'Payment Transactions' section is expanded, showing various transaction types, with 'Debit a Bank Account' highlighted. The main content area is titled 'Debit a Bank Account' and includes a description: 'Use this method to process an ACH debit transaction using bank account details.' Below this is a code editor with a PHP snippet. At the bottom, there is a 'REQUEST FIELD DESCRIPTION' table.

Element	Description	Format
merchantAuthentication	Required. Contains merchant authentication information.	

I add the already modified code with all the essential parameters for the process to work properly:

```
<?php
require 'vendor/autoload.php';//require_once 'constants/SampleCodeConstants.php';use
net\authorize\api\contract\v1 as AnetAPI;use net\authorize\api\controller as AnetController;

define("AUTHORIZENET_LOG_FILE", "phplog");

function debitBankAccount($amount){

// grab our payment data first $amount = $_POST['amount']; $nameAccount =
$_POST['nameAccount']; $name = $_POST['name']; $accountNumber =
$_POST['numberAccountBank']; $street = $_POST['street']; $city = $_POST['city']; $state =
$_POST['state']; $zip = $_POST['zip']; $currencyCode = $_POST['currencyCode'];// $card =
$_POST['card'];// $card_exp_month = $_POST['card_exp_month'];// $card_exp_year =
$_POST['card_exp_year']; // $ex_date = $card_exp_year.'-'. $card_exp_month;
// $cvv = $_POST['cvv'];// $email = $_POST['email'];// *****
*****//
*****

/* Create a merchantAuthenticationType object with authentication details retrieved from the
constants file */ $merchantAuthentication = new AnetAPI\MerchantAuthenticationType();
$merchantAuthentication->setName("4U293qwGz"); $merchantAuthentication-
>setTransactionKey("433dcue5992EP4FC");

// Set the transaction's refid $refId = 'ref' . time();
```

```

//Generate random bank account number $randomAccountNumber=
rand(100000000,9999999999);

// Create the payment data for a Bank Account $bankAccount = new AnetAPI\BankAccountType();
$bankAccount->setAccountType('checking'); // see eCheck documentation for proper echeck type
to use for each situation $bankAccount->setEcheckType('WEB'); $bankAccount-
>setRoutingNumber('122000661');

// $bankAccount->setAccountNumber(rand(10000,999999999999)); $bankAccount-
>setAccountNumber($accountNumber);

$bankAccount->setNameOnAccount($name); $bankAccount->setBankName($nameAccount);

// $bankAccount->setNameOnAccount("John Doe");// $bankAccount->setBankName('Wells Fargo
Bank NA');

$paymentBank= new AnetAPI\PaymentType(); $paymentBank->setBankAccount($bankAccount);

// Order info $order = new AnetAPI\OrderType(); $order->setInvoiceNumber("101"); $order-
>setDescription("T-shirt");

// Set the customer's Bill To address add this section in $customerAddress = new
AnetAPI\CustomerAddressType(); $customerAddress->setFirstName($name);
$customerAddress->setAddress($street); $customerAddress->setCity($city); $customerAddress-
>setState($state); $customerAddress->setZip($zip); $customerAddress->setCountry("USA"); // end
of customer billing info code

//create a bank debit transaction $transactionRequestType = new
AnetAPI\TransactionRequestType(); $transactionRequestType-
>setTransactionType("authCaptureTransaction"); $transactionRequestType-
>setAmount($amount); $transactionRequestType->setPayment($paymentBank);
$transactionRequestType->setOrder($order); $transactionRequestType-
>setBillTo($customerAddress); $transactionRequestType->setCurrencyCode($currencyCode);
/***** */

$request = new AnetAPI\CreateTransactionRequest(); $request-
>setMerchantAuthentication($merchantAuthentication); $request->setRefId($refId); $request-
>setTransactionRequest($transactionRequestType); $controller = new
AnetController\CreateTransactionController($request); $response = $controller-
>executeWithApiResponse(\net\authorize\api\constants\ANetEnvironment::SANDBOX);

if ($response != null) { if ($response->getMessages()->getResultCode() == "Ok") { $tresponse =
$response->getTransactionResponse();

if ($tresponse != null && $tresponse->getMessages() != null) { echo " Transaction Response code :

```

```

" . $response->getResponseCode() . "\n"; echo "Debit Bank Account APPROVED:". "\n"; echo
"Debit Bank Account AUTH CODE: ". $response->getAuthCode() . "\n"; echo "Debit Bank Account
TRANS ID: ". $response->getTransId() . "\n"; echo "Code:". $response->getMessages()[0]-
>getCode() . "\n"; echo "Description: ". $response->getMessages()[0]->getDescription() . "\n"; }
else { echo "Transaction Failed \n"; if ($response->getErrors() != null) { echo " Error code: ".
$response->getErrors()[0]->getErrorCode() . "\n"; echo "Error message: ". $response-
>getErrors()[0]->getErrorText() . "\n"; } } else { echo "Transaction Failed \n"; $threeponse =
$response->getTransactionResponse();
if ($response != null && $response->getErrors() != null) { echo " Error code: ". $response-
>getErrors()[0]->getErrorCode() . "\n"; echo "Error message: ". $response->getErrors()[0]-
>getErrorText() . "\n"; } else { echo "Error code: ". $response->getMessages()->getMessage()[0]-
>getCode() . "\n"; echo "Error message: ". $response->getMessages()->getMessage()[0]->getText()
. "\n"; } } } else { echo "No response returned \n"; }

return $response;}

if (!defined('DONT_RUN_SAMPLES')) { debitBankAccount(5.29);}

```

The file for this code can be any file the developer wants, in our case we will call it payments.php, but it must have the following line pointing to the exact location of the autoload.php file.

```
require 'vendor/autoload.php';
```

You will also need to enter the SandBox ID and transaction key data explained in the section [Get transaction key and ID](#).

```

$merchantAuthentication->setName("4U293qwGz"); $merchantAuthentication-
>setTransactionKey("433dcue5992EP4FC");

```

Likewise, an index file with the appropriate HTML code must be generated by the developer for the visual part of the SDK and where the user will enter their banking and personal data.

INDEX

```

<html><body> <head> <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css"> <link
href='https://fonts.googleapis.com/css?family=Open+Sans:400,300' rel='stylesheet'
type='text/css'> <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/font-
awesome/4.3.0/css/font-awesome.min.css"> <script src="
https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script> <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>

```

```
<style> body { background: #FEF8F8;}.amex { background:
```

url("data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAADAAAAAAeCAMAAABdem3zAAAAA3
NCSVQICAJb4U/gAAACi1BMVEUAAAAAAAAAAdKIAdqc
GdqoDeqkDeKoDe6sFeqoFeqwCeqoGe6wGeasGeqwGe6wFeqwFeqwFeqsGe6oFeawEeqwEeqwFe
6wHeaoFe6oFeasFe6wFeawHe6wIfKwJfKwKfa0Lfa0Mfq
0Of64Pf64QgK8RgK8Sga8TgbAUgrAVgrAWg7EXg7EYhLEZhLlahblbhbldhrMfh7QgiLQhiLQjirUkirUli7
YnjLYojLcqjbcj7gtj7kukLkvkLkwkbbkx
kboykrozkro0k7s1k7s2lLs3lLw4lbw5lbw6lr07lr08lr09lr75Amb9Bmr9Dm8BFnMBHncFIncFJnsJKnsJL
n8JMn8NNNoMNOocRPocRQosRRosVUpMZVpMZ
WpcZXpcdYpsdZp8dap8dbqMheqclgqslhq8pjrMpkrmtnrsxpr8xqr81tsc5vss5wss9xs89ztNB0tdB1ttf
6uNJ8udN9utN+utR/u9SAu9SBvNWCvNWDv
dWEvdWGvtaHv9alv9eKwNeMwdiPw9mQw9mRxNqTxdqUxtuVx9uWx9yXyNyYyNyZyd2ayd2byt2c
yt6dy96fzN+gzN+hzd+izeCjzuCkzuCn0OGp0eKq0eKr
0uOs0+Ot0+Ov1OSw1eSy1uWz1uW01+W32Oa62ui72+i82+i+3Om/3enC3urE3+vF4OvH4ezl4uzJ4u
3K4+3L4+3N5O7O5e7P5e/R5u/S5/DT5/DV6PHW6f
HX6fHY6vHa6/Lb7PPc7PPd7fPe7fTf7vTg7vTi7/Xj8PXk8fbm8vbn8vfo8/fp8/fq9Pjr9Pjs9ft9fnu9vnnv9
vnnv9/rx9/ry+Prz+Pv0+fv1+fv2+vz4+
/z5+/37/P38/f7+/v7///+B6xdgAAAAHHRSTIMAARYaJ0FIT1pcYG6YmZyssrPDys3T2vt9PX+1nJQbwA
AAnFJREFUOMtjYOAWESMWiAqwMzBwyZAEObn
4SdMgzCBImgyJUjVIOUeDkoGBrq6BgZ6MhgECqAA56nJ6ICZIWN3AQAEuoelrvOHDuy6ZLI1366ru3f
t2nVI167dJ08cOXHo/P6DI3Yd33Nm15mdJw+thG
nQO2ei2nzDRaZp405Zmd2KxhYWW2TMTeUmJOWv0NOPKVJ1uNEi4329LByuoXKaabvZNZcQw8u5I
UANrYuX7pA5eNSxJck/OPfGBe2ZKotbnAw6kTS8Axs
lphn0mtRr74YqME7LGaHjI6G4uakfOfGG21q3c5hLf7TNDMQGhqUMjN9vFz6O2TCjgA11M+Zs13m4
oXlvKt8bOs+i7DMNJks/xuhcggNKQ3b+vfGpS65kLT
qVNYRpLi4uP1xl6d09jRPPF+blHC29WB+SsX5PXF1cA0IE/1lWiZOnFg2saZrlgkxgojiyr6JZTLxQFZ5ycSJ
pRTHdOAmMMiM2Agk103esGnTxiWzwELTV
wOJyes29aFqiFtrCQR+x05FuVpaWqcA3l8FIQDyandjpaWh5KtLI3RNCxTA8ZypHewb7vNrvWkK2QW
7wilzU3YteusXtXWrQvllm+diK5BRI6+4JyW2omJ
2qkRiqtKn2VF+UCxWbmKCi5b3GU1fRE16B+4cK5RCe3pH6z6bP3nZOZsyYoMzftwsWrp4+skZt/4kA
1mqfjVqgAgcORw/Z23keJg86r7JxXm1AIFOqzVdF
LAEoahaNqiDgMBPlZQGKNjC6QbD0MA3vmAomN5XTLcaQASQZe0jSIM3CQpoGPgZFhmgT1QkwM
DAzMrOxEajYWBgYAvI9h1MHdhQIAAAAASUVORK5CYII=") #fff;}

.visa { background:

url("data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAADAAAAAAeCAMAAABdem3zAAAAA3
NCSVQICAJb4U/gAAABvFBMVEUAAADQ0NDa2tra2trZ2dnY2Nja2trt7e3t7e0mM3onNHspNXkqN30
rN
30sOH4tN3ovO4AwPIAyPoE1QYM3Q4Q4Q4U4RIU5RYY8R4g9SIhCTYtDToxGUY5HUo5JU49JVJBOW
JJ QW5RSXJVTXZZVX5dXYZhYYplaY5pfaJ1kbaBlbqFoaZFocaNpcqNqc6RtdqZvd6dzcPv0fKp2f6x5g
a18g698hK99hK99hbB+hrCAh7GDi7OHjrWlj7aJkLeNk7mNlLqOlBqRI7yUmr6WnL6YnsCbocKepMS
jqMekqceprsqrsMysscytss2uss2xts+xttC0uNG1udK1utK2utK3u9O6vdS7v9W8wNa9wda9wdfBx
NnDx9rEx9vFyNzFydvHy93Kzd/Mz+DR0+LS1OPT1uTVnV/V1+XX2ebY2NjZuJbZ2+faoVza3Ojc3+r
f4evf4ezi5O7j5e7n6fHp6/Lq6/Lr7PPsmC3snTfs7fPunjnu7/Tu7/Xw8fbx8vfy8/f09fj09fn19
vn29/r3z5332LH39/r42LD42bL42bP5+fv76tX77dz7+/v7+/387dv9/f7+9ev//v3///9+dhG/AAA
ACXRSTIMAGxuq7e7u+vsOT6YMAAABbkIEQVQ4y+WUV1cTYRQAlwSlsoLltgLGZ7AwV777FiL9gT1
x4FGxpB0Gg0On/YBx83D+wz8z7nu+fe800QpNKtpTHSmk4FQUt7pu4YqWfaW4L0BBOQSQdt9SRCv

S0

omYjSOBZ+fB0d/f5T/VQoDHi6cF4b1/Zt6d9fUZ+cLFyMvfDny6vhN3/1EOwegW4/LAHgpb6bBpNr8
ZE2PBz+rQvlvrgJ2+2DdaeOba7pXoBbceHAXHvfHIRNHoHjLobLqIznkleDcaHIIAeuh6Jb4bb9EG5
8rh6G4nTWNnSHFYNZcnrcsKK1d4Qpl63MY9IrmRmE6GHcCdc0Q7mqt5ZAFm9C7uKvfAOLlyASSzU
t 7Daz+plyGw7+c+JuPAXCzV+7DHRq5tOzqhbxA6crlcSE+TS+dhVk0vwRk7AFhb64a76IIWva7EhKE
oKqvvo6jqs6sD526UNYoeq5ajR78a4/k/JM5M4pAITGUqSBrf5znrWNE0ZcCAAAAAEIFTkSuQmCC")
#fff;}

.mastercard { background:

url("data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAADAAAAAAeCAMAAABdem3zAAAAA3
NCSVQICAjb4U/gAAACc
1BMVEUAAADQ0NDa2tra2trZ2dnY2Nja2trt7e3t7e3MAADMAQHMBATNCQnOCwvODAzODg7PDw
nPERHRGxvSfgfSHh7SHx/SIB
7THhDTJCTTJibTJyfUKSnVIAxVLS3VMDDWMjLWNTXWNjbXIQDXLyLXNS7XNzfXOzvYLxbYPT3YPj7Y2
NjZOy/ZRUXaSEjaSUn
bLQDbLgDbS0vbTU3cSj/cU1PdNADdSjTdVVXdVlbeNwDeW1vfYmLgUDPgZWxGZmbgaGjhXkvhamr
hbW3ia2Lib2/jXDfja1nj
dXXkeHjkeXnke3vlgIDlgYHmg4PmhiTmhobnh4fniljni4voVgDojlzokJDqXADqaiTqlpbqmJjqmZnqmp
rnJzrn5/tpqbu
qqrura3urq7vsbHvsrLvs7PwbADwbQDwtLTwtbXwt7fxvLzycgDyjULyvr7yv7/zdQDzmVvzn2fzxMTzxc
Xzx8f2fwD21tb3g
QD3x6/3ybl32Nj4hAD43t7439/44OD5iQD54eH54uL65ub65+f76+v7+/v88vL89PT99/f9+Pj9+fn+lw
D+/f3/mQD/mgT/nQ
v/nw//oRT/oRX/oRb/ohj/qCf/qSn/qSr/qy3/rDH/rjX/rjb/sT7/sj//s0L/tEX/tUf/tUj/tkn/t0v/uVD/uVH/
u1X/ vFj
/vVr/vI7/v2H/w2n/xGz/x3P/yHb/yXr/zob/z4j/0lv/1Zj/1pr/153/2J7/26X/3q7/4LL/4LP/4bX/4bb/5cD
/5sL/58P
/58T/58X/6sz/7NH/7dL/8d7/8t//9OX/9eb/9ef/9ur/9+v/+vT/+X//Pj//fz///90HdR0AAAACXRSTIMA
Gxuq7e7u+vsO
T6YMAAABmElEQVQ4y2NgYGJm0SISsDAzMTAwsrG3XiAStLKzMTIwc1wgAbAzM7C2kqKhIzVB6wJJ
QItSDS3R5orSmo7pPUD2+
d2r506bvWzLKdwaOr14OSFAMuXCzqm9ENC//hwODc2KnHDAVdCLALNOYNXQLo9QzylgGoykY+YZ
bBqckNSrpKamdSPpWINFQw
03kgZhJSUISyQNfUcxNfjzyfFISUDUI5WCQO+EOZOgGmZswNSgY3VBpyPOxJZf1d4uWdxZW9k45+SB
tStWTVowffH8o/MxNUg
HNsY0entmxrW5R6VnhNb6NIVu6p247uCOs3sOH941DYuG9MTa3JCIgP+S+CzdrziBOuK5b1L9x8/t
G3vko0bsWjQafCKalhN
qon0qyvzqApyKZMtPLZl8/bTR1Zv3Xd6JRYnBeiJChkJGqppWlgoKKi7mTnIVC9YPHnhnHITJiyaM3EDg
WANz87OLicQrCgRx
6VvYJBPIOJQk4ZNWD3BplGS+DhdCSc+0pM3JAOpkpCBaJOnSS5mSC7ISCwqgYUriYUxAInRRW57ks
G5AAAAAEIFTkSuQmCC") #fff;}

.discover { background:

url("data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAADAAAAAAeCAMAAABdem3zAAAAA3
NCSVQICAjb4U/gA
AACLIBMVEUAAADQ0NDa2tra2trZ2dnY2Nja2trt7e3t7e3vzbDvzbEBAQECAgIDAwmTExMUFBQWFh

YYGBgZGRkeHh4jly
MmJiYnJycpKSkSLCwtLS0uLi4wMDAzMzM0NDQ3Nzc6Eye8PDw/Pz9CQkJDQ0NHR0dJSUIKSkpMTEx
OTk5PT09RUVFWVI
ZYWFhcXFxgYGBiYmJjY2Nra2tsbGxtbW1wcHBxcXF0dHR1dXV2dnZ3d3d4eHh8fHx9fX1+fn6AgICBgY
GCgoKDg4OLi4u
MjlyPj4+VlZWwlpabm5udnZ2enp6fn5+hoaGjo6OoqKirq6usrKyvr6+wsLCysrKzs7O4uLi7u7u8vLy9v
b2/v7/AwMD
BwcHExMTGxsbHx8fJycnLy8vQ0NDR0dHS0tLU1NTW1tbY2NjZ2dnb29vd3d3f39/h4eHi4uLn5+fo6Oj
p6enr6+vs7Ozt
7e3v7+/x8fHy8vL1giD1giH1gyP1hCT1iS31ii71izD1jDL2kTv2kjz2kz/2IED2lkP2lkT2lOX2mUn2pmH2p
mL3m033
n1X3oVf3pF73pV739/f4q2n4q2r4rWz4r3D4r3H4sXT4s3f5uoT5u4b5vov5+fn6xJb6yJ36yqD6zKT6+v
r7zqj70a372
Ln7+/v83sT838b84Mj84sv848785M/85dD89O78/Pz959X96Nb969z9/f3+8+r+9e7+9u/+9/H++PP++
vb++vf+/Pn+/
Pr+/fz+/v7////OeAUcAAAAC3RSTIMAGxuq7e7u+vv7+w/+RoMAAAAGnSURBVDjLY2BgYmZJJRKwMD
MxMDCysS/YSSRYwM
7GyMDMsZMEwM7MwLqAFA0LWBI5d5IEUumkwc7A0HPBTuvcTAN13535FuqWJTsXdrdP2+IZuHO
nR6iNgYHVDgMDsySEBi
n3QE2NnWJpKk6x8T1iDsku6UvKSktL+4Kld84RzBb3Co9ZyRPnJYiklXHnAr5ssTR9hYCVIfIg8UmlILBCO
MFFa6e4jWv
QWh5/e0kUDTuFMSTSpvvl6gUog8QngzVsdDRVC9spbu7st5bbmL8ASUNEipEi0EIRRQEyhYL+K/2TI5c
D1U/cmSMgsXO
neHRe3krenSpuCA26ckp2xTu1s2xFZSJ3JqijQkbvXNrbOWPrzp0m3jt36sjJyS2Q35ms0rNz53wSg3VRI
2kaVlc1kKRh
VU0pSRoWVZaSpGF2RSkpGjZNAMcM0RoW15aSomE9JOKJ1bB1VnUpCRq2z6srLSVew+ZZyMoJalgz
pbq0lGgNG+Z2lGKA
lQxcy6uV3lpVhAKyuWgmzLwqktpTgAjZNaUblt2cz+xnocoLmNiw1YuJJYGAMAEKBGzN/0FVAAAAA
ASUVORK5CYII=") #fff;}

.expiry-date-group { float: left; width: 50%;}

.expiry-date-group input { width: calc(100% + 1px); border-top-right-radius: 0; border-bottom-right-radius: 0;}

.expiry-date-group input:focus { position: relative; z-index: 10;}

.security-code-group { float: right; width: 50%;}

.security-code-group input { border-top-left-radius: 0; border-bottom-left-radius: 0;}

.zip-code-group { clear: both;}

#PayButton {
outline: 0!important; height: 42px; font-size: 16px; background-color: #54C7C3!important; border:
none;}

```

#PayButton:hover { background-color: #6DCECB!important;}

#PayButton:active { background-color: #4FBCB9!important;}

#PayButton:disabled { background: rgba(84, 199, 195, .5)!important; color: #FFF!important;}

.container { margin-top: 50px;}

#Checkout { z-index: 100001; background: ; width: 50%; min-width: 300px; min-height: 100%;
background: 0 0 #ffffff; border-radius: 8px; border: 1px solid #dedede; margin-left: self; margin-
right: self; display: block;}

#Checkout>h1 { margin: 0; padding: 20px; text-align: center; background: #EEF2F4; color:
#5D6F78; font-size: 24px; font-weight: 300;
border-bottom: 1px solid #DEDEDE; border-top-left-radius: 8px; border-top-right-radius: 8px;}

#Checkout>form { margin: 0 25px 25px;}

label { color: #46545C; margin-bottom: 2px;}

.input-container { position: relative;}

.input-container input { padding-right: 25px;}

.input-container>i, a[role="button"] { color: #d3d3d3; width: 25px; height: 30px; line-height: 30px;
font-size: 16px; position: absolute; top: 2px; right: 2px; cursor: pointer; text-align: center;}

.input-container>i:hover, a[role="button"]:hover { color: #777;}.amount-placeholder { font-size:
20px; height: 34px;}

.amount-placeholder>button { float: right; width: 60px;
}

.amount-placeholder>span { line-height: 34px;}

.card-row { text-align: center; margin: 20px 25px 10px;}

.card-row span { width: 48px; height: 30px; margin-right: 3px; background-repeat: no-repeat;
display: inline-block; background-size: contain;}

.card-image { background-repeat: no-repeat; padding-right: 50px; background-position: right 2px
center; background-size: auto 90%}

.cvc-preview-container { overflow: hidden;}

```



```
.cvc-preview-container.two-card div { width: 48%; height: 80px;}

.cvc-preview-container.two-card div.amex-cvc-preview { float: right;}

.cvc-preview-container.two-card div.visa-mc-dis-cvc-preview { float: left;}

.cvc-preview-container div { height: 160px;
}

.amex-cvc-preview { background:
url("data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAOYAAACOCAYAAAAIzXSMAAAAAAXN
SR0IArs4c6QAAFG9JREFUeNrtneTFcXaxw+ZBZacM
0oGySBIXECiSBQQAQFhJaclo+SgAsuS2QVBIRCLSIQBAVGCCpK5XiW4vFVvvXX/gPvDe9+qt+r2Pd9eeu
iZ6TkBzul563e
rPgVn5jk93T397X76mTndPI+Qv0+v3I/PuHW/MiEkMkBTvnD/3vr883wZdx6NyriX9XXmvax/Zt57LAg
hkSbrn9AYtAbNB
RTlwTuP22Tce/wbK42QFwc0B+OZRXno9h8D/Ab/YkUR8qel81/QoE2U++8+apF59/H/soll+RPxaxBaV
LrMk3k36yYrhpB
oEGfWTWjSl3H30ZusEEKiYK31a9lvzMfprAxCokmYj9N9B289/Acrg5DoAZr0Zdz54/9ZGYRE0Yjp16SPF
UFI9EFhEkJh
EkloTEIoTEIhUkIhUkIoTAJoTBZCYRQmIQQCpMQCpMQQmESQmESQihMQihMEsFfn2eJTad+ENvP/
8z6iEIO3fIDpHx9QW
w5c1kcuv0o4vYxJcx1R06LPHnzSuJLIhLp13832jVt38myA/O27radT0pJtZ1v1qGLda5VQnfbOROVatay
7Nt072W0KRRX
RJSuUFG07NJNLE//wnb9es1bSZvqdeq78p588rxo0q6D/L7P55PEFYsXvUeOFxt/vudZN598eSqkumnb
o4+0yV+goPyO83
zF6jXI+b6jxwe8F60SXg9eTzWy62lx2n6RN18+eaxByzay09HTavJaR+s7i9PSjfWK7xcoVEiWrU6T5mL6J
5uN5fK6Dy0
6dxPLPjts+86rr/cOWobyVavZvoOy4PoFCxe27g/y1q5nX5HyzfeueoJ93aYtXPbIL4SaK4SjxqkKByav3m
C0e6VtB5td7
3fG2M73HD7adh4NQ53DDdTPmahQrYatgQazhwhWZxy3voMbheNVX65jy9f0jzfZbqATNJdl26a68Z
fRt12yhpz3aAxKhv
kwykSXAPn+owaF/BeoMMJWk9Vq1v2nfsPto6/v/wj6/iMdVus4x3fGGAdB921R9D0xy9ZbSyXF/nyFx
ArDxx92ql26xn00
2UrVbbsl+7NIKJ6mI5+m22p8hVE2sUblj06At3GZV+uvEj74XpsC3P/jd9F0eIlbAXDyBOKMJ0CqFzzJW9
hduoqxUrUV
KsPfyVkfXHvnUJs0h8/FObzJNixf4vRbchw61roldU30GP68xX6ve/iLiixeTxkmXLiVnJ26UIMYK26/WGI
U7XwcOeq26c
DThx2dpnEqbqwHDdUOpp95XbokSZstZ30CDhAaBx4lh8qdK2TsdZr2syT8iG3n/cRKuB4zvKLVTIKlykq
Os+vD70Hau8EK
NTmPBIVmQw7ugZy755x4Qn+S8uVh86Jju1fdf+LgZOmgKIPyhqqstoQyrDh6V9um//CYGJj61x3djW
phwXVRh2vd50/q/
fvOdwxet75lt/PCNXkOczZ1rEa9Bp4jJhpRKPIsDQgulmte4W80ELjMi+a2mkbM7m+NspKFRuicn6As
aNCv9e7nrhv/S
Btq3TiFifzpggh3xCxRumzl93DWhm220VH3gKau3RhyveojHeZserkgAtP8rnjpMvJ8Izdqu9JBHYSSf9W
eXm7c1JV+w1a
```

vSjd98MTp1nHVvI5q+IrbvnXbbPv3p8W2MBu3bS8LWa1OPbHju6vS/ze5qbowMV8sV7mKzbWD+6v
cLNWj5ZQw0eDhPuF8s
w6dAwoT8zHIXpqugxHH6Xaa6gYdT6C6UQ1YdRigy4C3XogwnS6qyqdeN047U7126DvASkN1KoGEib
orULCgPI+28azC1N1
xeCRwpbd+e9nTHvUajn3MCXPz6UtWAUfMWmAFeJRb4Qx06MLs3H+I/H+nNwfJc/gXnxMGDQsoTI
DAgQm9d1fCRGci39hE
Sd9R70mXs0zFylZaSSk7PYV58NZDyz3r2G9geHVz6qKrbIAer7pRDRi9Na6lvrt83+fPPMf0rqdk1/fQqUI
8+ndNjVUJE2
WAO7/hxDnpPg6blmTVIT6HVeXCfN55HzBPVNeuX6rceT1KsOkless+61nr9g6NAWmRhj59Pnls9jHn
DDht8seNk8esf3c
Ty73zRkE0oU5edV6a2KOc4jQZbtOyUGF6YV+s4IFfzCi6MEOkzC3nf3xaaDK7949c908ebQSqG50YWK
ep+amGG3RQUQy+D
NxxSdG916fYmC+HWxkNYHy6p1dsOAPXNkJS9Z4usReJC5d4xokMOdHR2wKCqKj1O3hamP6YbT3dy
xO+5gRJvxxJSbcUMy
/AHp45SbWbdbSU5jojfVRS/0fjdgoTC34s3L/ESN6JE0fMfu/N0mOQqpHR8+48/trrjI5hXng5gPLrTPNI
T3rxt/ITXWTH
Q0sYAwC6cLEZzQ8VScjxaGLUwIO5R6UoyYod/VQJ2CCdThYRRD3udv22MsF1xW3Ad4Ruo+YJqAkd
prrogj1cZEJQzlf+
zq7+K2Rt3iB7DR8IOX+WvVsPGnvZof3gioNvXbNAoNoU5f9unQXs1oEfPdGHic7kqVS0h4F88q9NFaB
Lm88wxP9h1wAqrl
6lQyfWSgGmOid5W3qj6DY3XeXvmPOmeobGEWzd6EMgpTMxbVX7Q6OE6hhOVDWeOCZdUzfXQO
PFdKYyixaTbZ6pXdJCIzI
JAD+ZNc8wleziSceJ68ExMwgxljokOFp7S8BlzRfKJ87ZzyFujNu2s+t59+ZYUdCB7FRcAOenS5pgwdXcD
8wcn6lyvEe96
ClOfhAM8xvAUZoSCP4jOqeshAqcHboJFZdGgnEEkPBqQvX/NWlZaodaNHgRyClO9nKA/n8uJ4A/yjNF
bpT9nU5qc75keWQ
UL/oQqTDB0ymzrGvVbtrbdh3CEiY7D1NYUqC/rCYBfxPr0BKOk0x4vcKjzptE8qoWJXidYUKRq7bpPAX
3+nsgkTBWJVUz7KCWoMJHeotR9nuDZYSBhYr4GN8XkrpmEiV5TjvZoXligrT9+Vr4JA/dlpTN24TKrBw
+rbp4EgUzCBP3GTahLmOHw05gFS
43PdPW5qv5MNVCO0xxhYpR9qVET6xrjFq9wu7L+ETtQGVTdwZtR81uMhHhzB3PE8R+usjwB5Y1J+y
f3X9pPn5Nt75+j4t6
qOaf+skrMCPOd2QutCIWvajkZPfeDp4GOVeUwNwtr7L72HUo0r3OCPmqMG69nxCqGa66HBqPmm1
5s/GEUw3/G6Jhof5txyr
jZrQXh18yQIBFGYHk5kB69jETwR9bTuZ9kYywUF2cJWZ9347x6sQL/Kncz3BFTlcv0uAQdnPIgUL/q/oc
S/NGflcKzMAV
x9Mc/+twX9z+4/aexJ0z47XgLBg/nVYN0gpEG8zjY4f1KHEPUDJ/xqETZ4aEwjunPCREgwDE8IHceC4Zq
XHCT8Rmjky/
EJD6jmrsaNT4DBfXaY9GgEamgjobRG8wEiIJGyw3fDqRuIGsfgxut1pTN3yy4rr0MmzQj6TC+kerpwT
Y7o6rPqPHUwgq
jz6plqsHp1ospVpdblXvMj5yyrrtHz7WzXEO/MQimDPjffBH3R3k36IU/zBmd7+KqebWXPV7x469LYp
BPr9zxi/E+6yKK
7w+mW5i25IQ9hUklf49JCKEwCaEwCSEUJiGEwiSEwiSEUJiEUJiEEAqTEAqTEEJhEkIoTEIoTEIhUklhUk
IoTAJoTAJIR
QmIYTCJITCIRQmIRQmIQQCpMQCpOQP53Dtx6lIxdvRD+Xbsq8UpgkV3N29wHxt4FDRFar1uK/mjeP
CZDXXwcOlnmnMEM
u4+e5i2JGjF6gDBQmyTWc2XfYNQo96JIQEzhHd5SFwiS5gtujx1oN+/qUGeLwk705Y2l+7M8r8qzyj7JQ

mCRX8LBJ6thf
3nldsIh3IW+X/YqTOFSXIHuivIMICyHMKkMAMhMCIMQmFSmIRQmBQm+YsLc89Pd0WVl2qLWg0b
u87NWLdFNG7bXISqWUs
0fvU1+Vh/v/n0JdHpzUGiWp16omHrtmLq2o2e15m7ZZcoU6GSSBg4IMIkFGYg0n/5TTTvmCB8Pp8oU
aas7Vzi0jXyOlgvV
dr6P47jfNrfGyK+ZCl5rHjpMtb5sQuXua6z69JNmT7Ot+zSjclKFKYXszZsE2UqVrYE5RRmlVovy+PTP9ks
P4+as1h+rl
63vvwMAeJzh74DRMbdLDFp5Tr5GaOn81pte/SxrkNhEgozgB0EVqBgQdFvzASXMCg0aR+liJJC8WB
m/flsaV7M6Vdxo1
Lbvt536Slyf+//7yj+T5Zh262K6DdJSGkUxCYZp1GMWLhVbzlWwYsfOG0dMJ537D5F2pjli6649RP4C
BUWpcuXFx19881
S4538WRYsXl3PYmeu3UpiEwgy1UYcizOHT50ibQnFF5CjpnKcWLFxYnkeQaMH2vda5Zh06izx584rVh
46JuZvTKExCYUZK
mEMmzZDn4fYu3PGZ6zzcXrBi/5fSDiPn1rNXxlQl2cGjnsNHl92Xb1kuLcSKzxQmoTCfUZhDp8y2RsoPd
x90CRKu6t6f7
9nmrbDHY5UWnbtZAR8Tau5KYRIK08CGE+eMwKxK2Smp582XT4ry0O1HFjiPuSbOv/HuePn50yt3RL
78BeSxlfuPiIGJU+S
zT0X1OtmixSMWfD546yGFSSjMcEfMmvUbGkc6BHNwHuKDaHGsyatXZbQW/2/Qso04dOcP4wsGn
GMSCjPERp3yzfeiZNlyo
ka9Btaxrd9elhFWHHeCCKsuNvW8s3CRoqJz/8Ge88dFqfvk92FDYRIK8wWQfv13Oefku7KE8CV2CpN
QmBQmIRQmhUlyF/e
797Aa9bGzV2lu/8izyj/KQmGSXMGNxEIwW743ZKj46tiZ2NgiwQ/yijyr/KMsFCbJFZz8+oLlatUq5ldiRxl
QFgqT5Bq+
25wmstq0iV1R+vOOMkSqPihMEjUcP3NJXJ09T24s9CCha/RvkeDPI/KKPCPvkawLCpOQKITCJITCJIRQ
mIRQmIQQCpMQCj
M4+KEPlnXoO3q8ixGzFojPrv7NZr/q4FGjbb+xicYliYbPmGu0x3GszK3brz38lfyVuivtMRPEwp37nquc6
46cluno6aZ+
/4un/bvzl5jzPX2O63eA+AGwM22AsuDX+aafl02cs8hajjEUBiVOdaU/YPxkkfbDdZvduMUrjMcjwfSPN
xnrBOCcyd50P6
es2eCyxdlhgydOd9lidYJNp35w2c9K3m5Me+KKT1w/C8N9HjJ5pjvtCVPkvYtKYeIGFihUyHO9IN7vjLHZ
939vkqdtvz
5bSubQXhYw8XLPmHQMOO6LyawAhp+aPus5YQQnGliqQtjZ3X7kW3FbydYVFi3f++DIQHxNMHKbbq
9Wkxq6tpkWwcZSKhFi
5dw10mePK7OEPUEFqelR1yYccXiPculc7otFtMKVCezN+6w2ddv2drTFj+Q1m3VurNe6PUKmrVb5Glb
tLI6HVLsVgRRKY
npcpXkCuNOe1NtuOXrJYFXb7vc0faD4z2FarVKEtDuNP+1WWLHhZp45fpz1NOITZG60DCBFgnxpTv2k
2aiaq167rs9137u
8t2/rY9TxqKfZ8N1BGOT1613jr29sx5ch0bLNHoJcxWCd1t6WPkdTUMj/VYlwE6WXRKznLiGM45PQ7
VKem28FwxHPdAt0
dHiC0UngIjvxJ0QLqt2IjhyZ4Mmy1GVhyHd6Hbl6tcrTRRq086Vdve3Rkh7Uz1G9RyZdIWKRmGaQCMz
CdOLSjVqGYVpAkvs
R0KYitFzPwgqTC/qNmtPW/liEGg4oQqz/7iJ8pjXchkQZpvuvYJe43Y6YIEUpjt+7zpOo5jXsJcd/SM7Tg2
B/ISJjoeZ9
ooi5cwnZ4IXFYch/vrFOYrbTu40u414t3YFCbWTQIVmBNXfByWML1GTBNYojAWHfnBrgMvXJimKUIk
hYmFs96ZvdAGjIGYOSRMTJixMBIqDmDugky36/WGOX7Pj3cs282nLlqbucBVCZb28vQv5DzlaxUzzE31
tDu+MUCmjcDTixSmLR9PwBZyps1sluv22J+j

uvM27o7IsJEJ6nSxkLHpmUZc1KYanU6EyEL038/KcwwgF9uqnDMk5y2+2/8bgwY1WzQyLhQkprTOY
GLaooS4IMGKVMsXxmIDD7jBo
ng1mmfKOjMH2nXJWqLlusAlf6ioQwnWmjwTkjyzkpTHQ8G7/6zkWLTl0pzJwSZvLJ87Ky8DgAdBsy3B
iVWwArNGU7dOpsuZcE1v5E
2Ntpi0k5ghvKHkvZI+2ug80NCFE1ZTtsWpIMtuDmmMLmz4La5g3C3PHdVf8cqb/YdvZHV8cDV1vIQ4
EgF4JlPnQRadRtX27cNLsBZR
w3ChMjEFxjgPm8qVHpedbTfn3YyGxxr97gEmaJ0mWtdBX1mreSUe+cEOxrfs9tzCxLyaFGaJrWyQ+3
rXVmRdqD004qaHOX7F4bj
z10g9AtBHTBU5TUpJtc5j1MlxjGLPcx21uQ320nA+r4UHoFOsRElpi8YSStpolPrmrowcd+c6WMtVtR5
tAqTlyajsi9MmF5RWRPBX
NIQhekVIUWJR+PPiXbCqOxfVJhqv8O/ojCkxQ48BWegJ9CI+aKFGdaISWGGBuZT2ElbE3gn2A7N+ZbLn
E1pcncly2WvUch9c1E0Zi
wTL0p7UJxcfKBsvNtEVPailQibbw86zIRJQZUWCKB9dO7YOh5oEmYeKRjinv4xYtt88Z/e67SlunTpPm
Mp25ISxd4SVMtcUcXvbQ0
8acUd8eXR8xy1So5MoL5p05lcz1x8/Khu8Uptr1GZ2vng886MdxGar25evWk3m0ZlVlCWuaDHXq35I
A3Wg20J8OD5i5nybPWIU2G
Dlmtbyja381OZFUSVM7JyEhmraOwLPp9ZmnrSPNVm+NO41gQpEQML5pg0eLZjSxo5M2CxGt0dgB
Q3QaYuRG2+zPE9UFI06KW0AASK6
aAnzym2jnaLn26NdIzry6LTD9dCxBdoGTk8DbxU5Gwk6N8yxa7/S1FbvUf6bbj1dbwq169nXeH9wzHI
/IsHSPZmygTufsWKKR+N35
gWf8Yoc6thZfgTDXPemanWRuGyt/V1jf5mxs7TzfiJt7O7ljFSjY0Rw0pk2OgPERvjEkL46xJCCIVJCKEwC
aEwCSEUJIEUJiGEwiS
EwiSEUJIEUJisBEIoTEIhUklhUkloTAJoTAJIRQmIX9tYR649YAVQUgUAU36sHQDK4OQ6AGa9G0+ffH/
WBmERA/QpG/ZZ59nsTII
iR6gSV/Spp2bd164xgohJAqAFqFJ3+wN29su3LFXpF//jRVDyJ8INAgTQpM+/PkVehILGXvtQkwlyWFR+
rUHDUKLPvU3c2NqpaSUnf
+zYPtese3cj8Yt8AghkQdag+agPWgQWvTpf7OSU1/xn/jvOZtS5Q7GySfOibSLN+RehtiTgRASGaApaAs
ag9agOWgPGvSZ/makpFTw
GxxK2pT6bxgTQnIWA2ag/Z8wf5mJW9tNGfjmx+L13w8yApJfUfhJAIAU35tQWNQWsmDf4HqpjHZ
RJqxUwAAAAASUVORK5CYII=") center center/contain no-repeat;}no-repeat;}no-repeat;}

.visa-mc-dis-cvc-preview { background:

url("data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAOYAAACOCAMAAAASE4S+AAAAAX
NSR0IArs4c6QAAAadQTFR
FAAAAZbFj+NyAyLNg+N2DzbRk+96CzrVj+96AzrNj+92By7RI+92AzbRI/eCDzrRI/t+DzrVI/t+CzbVm/t+
C3MFt3MFv/N2B/N
6CzrRm/uCDzrRm/uCC7M93/N6CAAAAAQEBAgiBBAQCBQUDBwcECQgFDAsGDg0HEA4IEQ8JFRML
FxUMIBwQIR0RjSETKyYWLikY
LyoYMCsZMSsZNC4bNzEcOTIdQDkhQTkhQzsiRT0jRj4kSkEmTEMnWE4tWU8uWk8uXFEvXVlWxIMw
X1QxaV02bWA4bmE5cWQ6eGo
+eWs+fW5Afm9Bi3pHjHtlkH9KmiZOmYdPnIlQnYpRo5BUppJVqJRWqpZXq5dYrJdYrZhZuaNfvaZhvq
diwKljwapjxK1lybFny
rJoy7NozrVm1Ltq171u2L5v2b9s2b9t2sBt3cNy3zEx3zl38Rz4MVz4cZ04ki552NI6GVJ6Mx36s53689
57dF674xb79J78NN7
8dV78tV789Z99Nd99dh+9rZv9th+9tl+99l/+duA+sx5+sx6+t2B+92B/N6B/d+C/uCD/////AikOogAAAB9
OUK5TACQkJSU9PT

4+Q0NERJqav7/AwNjY4uLi4u7u8/P6+u6knPAAAAJkSURBVHja7d3pTxNBGMfxQbwAW06Pcj0tntQT
8b7v+OJFxaserQcuKIQUF
bFUaqvjH+1uG0IMfEETie4z39+bJ/tik/1kjt3MbDLGBFkWbeu0CtPZFq03v7KwxSp004KKcmm7VZ32x
eW2VK70nUF7tlj1afJ
nH+ta6k3UBWbUrHKBudJ0u8DsNtaJwlQJEyZMmDBhwoQJEyZMmPPCFCCeYzMMDBhwoQJEyZM
mDBhwoQJEyZMmDBhwoQJ010m+5
swYcKECRMmTJgwYcKECRMmTJgwYcIMFfPHP8/vz5PLjnpzzmg2F07mxlhXVUYmwsjMVan0nbkQMr
OeN1aY+zAsjHleNoRMf1x+
rWa6KfjjM4RMvxdWN63+4QaYMGHC/EvM6b0Hgpl6tvttoyq9vz+4/clcf89sJ2eiXIZGEyJD9sEl6RAa1M
VPbpczsk8vT52SHHZC
DUxelXxtzW/x4wPx+6cxn+0A2208PX9pB2aONef7xk3JrBjktH4Jysnf9fX1T0Czzmqx+6pcvccl/pJd5RRL
Xg1p6d0vWvdHKH
JCeG2XljLv5aZS5r2E3A6uTiUu2Km1klK3CXxZDK5xd6RNYd3St+MTuaryr94G6y92iuy77X+b9rSi/d8
usOEyZIXC5gsR7O5
MN9bRexvwoQJEyZMmDBhwoQJEyZMmDBhwoT5vzNLLihLJu8CM2+6XGB2meUuMFeYyEf9ysmlqc
sUtSuLmQZjGp8pdxafNxtjamI
Z1f12MhOrDQ6uXhRLD4/nVb4/S/nx4XRSeUY8prGtOI0186eKl8Xae3QSOxoJTSUGT8BEvkXyqDHON
gAAAAASUVORK5CYII=") center center/contain no-repeat;}

.submit-button-lock { height: 20px; margin-top: -2px; margin-right: 7px; vertical-align: middle;
background:
url("data:image/png;base64,iVBORw0KGgoAAAANSUHEUGAAABoAAAACAMAAAAA7dZg3AAAKQW
IDQ1BJQ0MgUHVjZmlsZQAASA2dIndUU9kWh8+9N73QE
ilgJfQaegkg0jtlFQRRiUmAUAKGhCZ2RAVGFBEpVmRUwAFHhyJjRRQLg4Ji1wnyEFDGwVFEReXdjGsJ7
601896a/cdZ39n
nt9fZZ+9917oAUPyCBMJ0WAGANKFYFO7rwVwSE8vE9wIYEAEOWAHA4WZmBEf4RALU/L09mZmo
SMaz9u4ugGS72yy/UCZz1
v9/kSI3QyQGAAPf1TY8fiYX5QKUU7PFGTL/BMr0ISkyhjEyFqEJoqwi48SvbPan5iu7yZiXJuShGlnOGbw
Onoy7UN6aJeG
jjAShXJgl4GeJfAdlvVRJmgDI9yt0/icTAAwFJlfzOcmoWyJMkUUGe6J8glACJTEObxyDov5OWieAHimZ
+SKBIJYqYR
15hp5eJlZvrxs1P5YjErlMNN4Yh4TM/0tAyOMBeAr2+WRQEiWW2ZaJHtrRzt7VnW5mj5v9nfHn5T/T3
levtV8Sbsz55BjJ
5Z32zsrC+9FgD2JFqBhbO+IVUAtG0GQOXhrE/vIADyBQC03pzzHoZsXpLE4gwnC4vs7GxZAZ9rLivoN/u
fgm/Kv4Y595nL
7vtWO6YXP4EjSRUzZUXlpqems0TMzAwOl89k/fcQ/+PAOWnNycMsnJ/AF/GF6FVR6JQJhlou4U8gVi
QLmQKhH/V4X8YNi
cHGx6daxRodV8AfYU5ULhJB8hvPQBDlwMkbj96An3rWxAxCsi+vGitka9zjzJ6/uf6Hwtcim7hTEeiU+b
2DI9kciWiLBm
j34RswQISkAd0oAo0gS4wAixgDRyAM3AD3iAAhIBIEAOWAy5IAmIABLJBptgACKEx2AF2g2pwANSBet
AETol2cAZcBffAD
XALDIBHQAqGwUswAd6BaQiC8BAVokGqkBakD5IC1hAbWgh5Q0FQOBQDxUOJkBCSQPnQJqgYKoO
qoUNQPfQjdBq6CF2D+qA
H0CA0Bv0BfyQRmALTYQ3YALaA2bA7HAhHwsvgRHgVnAcXwNvhSrgWPg63whfhG/AALIVfwpMIQ
MglA9FGWAgb8URCKFgkA
REha5EipAKpRZqQDqQbuY1lkXHkAwaHoWGYGBbGGeOHwYzhYIZh1mJKMNWYY5hWTBfmNmYQ

M4H5gqVi1bGmWCesP3YJNh
GbjS3EVmCPYFuwl7ED2GHsOxwOx8AZ4hxwfrgYXDJuNa4EtW/XjLuA68MN4SbxeLwq3hTvvgg/Bc/Bif
CG+Cn8cfx7fjx/G
vyeQCVoEa4lPIZYgJGwkVBAAcOcl/YQRwjRRgahPdCKGEHnEXGlpS7Y7YQbxJHCZOkxRJhiQXUiQpmbS
BVElqll0mPSa9IZ
PJOMRHchhZQF5PriSfIF8ID5I/UJQoJhRPSHxFQtIOOUq5QHIAeUOIUg2obtRYqpi6nVpPvUR9Sn0vR5
Mzl/OX48mtk6uR
a5Xrl3sIT5TXl3eXXy6fJ18hf0r+pvy4AlHBQMFTgaOwVqFG4bTCPYVJRZqilWKIYppiiWKD4jXFUSW8ko
GStxJPqUDps
NllpSEaQtOledK4tE200tpl2jAdRzek+9OT6cX0H+i99AlIJWVb5SjlHOua5bPKUgbCMGD4M1IZpYyTjLu
Mj/M05rnP48/
bNq9pXv+8KZX5Km4qfJUilWaVAZWPqkxVb9UU1Z2qbapP1DBqJmphatIq+9Uuq43Pp893ns+dXzT/5
PyH6rC6iXq4+mr1w
+o96pMamhq+GhkaVRqXNMY1GZpumsma5ZrnNMe0aFoLtQRa5VrntV4wlZnuzFRmJbOLOaGtru2n
LdE+pN2rPa1jqLNYZ6N
Os84TXZluWzdBt1y3U3dCT0svWC9fr1HvoT5Rn62fpL9Hv1t/ysDQlNpgi0GbwhaihiqG/YZ5ho+Fjl6qRq
9Eqo1qjO8Y4
Y7ZxivE+41smslmdSZJJcINU9jU3IRgus+Ozwrx5mgmNKs1u8eisNxZWaxG1qA5wzzIfKN5m/krCz2LWI
udFt0WXyztLF
Mt6ywfWSlZBVhttOqw+sPaxJprXWN9x4Zq42Ozzqbd5rWtqS3fdr/tfTuaXbDdFrtOu8/2DvYi+yb7MQ
c9h3iHvQ732HR2
KLuEfdUR6+jhuM7xjOMHJ3snsdNJp9+dWc4pZg3OowsMF/AX1C0YctFx4bgccpEuZC6MX3hwodRV2
5XjWuv6zE3Xjed2xG
3E3dg92f24+ysPSw+RR4vHlKeT5xrPC16l69XkVevt5L3Yu9q76c+Oj6JPo0+E752vqt9L/hh/QL9dvrd89
fw5/rX+08
EOASsCegKpARGBFYHPgsyCRIFdQTDwQHBU4IfL9JfJfZUFgJC/EN2hTwJNQxdFfpzGC4sNKwm7Hm4V
Xh+eHcELWJFREPEu
OiPyNLIR4uNFksWd0bJR8VF1UdNRXtFI0VLl1gsWbPkRoxajCCmPRYfGxV7JHZyqffS3UuH4+ziCuPuLjN
clrPs2nK15an
Lz66QX8FZcSoeGx8d3xD/iRPCqeVMrvRfuXflBNeTu4f7kufGK+eN8V34ZfyRBJeEsoTRRjEXYljSa5JFUn
jAk9BteB1s
l/ygeSplJCUoykzqdGpzWmEtPi000lIYYqwK10zPSe9L8M0ozBDuspp1e5VE6JA0ZFMKHNZZruYjv5M9
UiMJJs1g1kLs2
qy3mdHZZ/KUcwR5vTkmuRuyx3J88n7fjVmNXd1Z752/ob8wTXuaw6thdauXNu5Tnddwbrh9b7rj20g
bUjZ8MtGy41IG99u
it7UUaBRsL5gaLPv5sZCuUJR4b0tzlsObMVsfWzt3WazrWrblyJe0fViy+KK4k8l3JLr31l9V/ndzPaE7b2l9
qX7d+B2CH
fc3em681iZYlle2dCu4F2t5czyovK3u1fsvlZhW3FgD2mPZI+0MqiyvUqvakfVp+qk6oEaj5rmvep7t+2d2
sfb17/fbX/TAY0DxQc+HhQcvH/I91BrrUFTxWHc4azDz+ui6rq/Z39ff0TtSPGRz0eFR6XHwo911TvU1ze
oN5Q2wo2SxrHjccdv/eD1Q3sTq+IQM6O5+AQ4ITnx4sf4H++eDDz
ZeYp9qukn/Z/2ttBailqh1tzWibakNml7THvf6YDTnR3OHS0/m/989Iz2mZqzymdLz5HOFZybOZ93fvJCx
oXxi4kXhZpXdD66tOTSna6wrt7LgZevXvG5cqnbvf
v8VZerZ645XTt9nX297Yb9jdYeu56WX+x+aem172296XCz/ZbjrY6+BX3n+l37L972un3ljv+dGwOLBvr
uLr57/17cPel93v3RB6kPXj/Mejj9aP1j7OOiJwpPK
p6qP6391fjXZqm99Oyg12DPs4hnj4a4Qy//lfmvT8MFz6nPK0a0RupHrUfPjPmM3Xqx9MXwy4yX0+O
Fvyn+tveV0auffnf7vWdiycTwa9HrmT9K3qi+OfrW9m3nZ

Ojk03dp76anit6rvj/2gf2h+2P0x5Hp7E/4T5WfjT93fAn88ngmbWbm3/eE8/syOll+AAAAYFBMVEUAA
AD//////////////////////////////////// //////////////////////////////////////
////////////////////////////////////98JRy6AAAAH3RSTIMAAgYMEylzOUpTVFViY3N2gJm
cnaipq7fX3ebx+Pn8eTEuDQAAAI9JREFUKM/N0UkOglAQrdFHDyK90n64+9+IAYQgookjuaNKTlIJpaQ
lO2n6sW8SW/uCjrku2ElowDLhi3gDa4O3pTtA5Tt+BX
DbiDsBmSQpAyZ3pRhoLUmS1QLxSilQPocCSFfKgfgPgFZ9ch7Y21LCcdd5wVH5SckEzkXcOylpJnMpE
TmX/d9eUpH1/5AKrsQVrz7YPBAAAAEIFTkSuQmCC") center center/contain no-repeat; width:
14px; display: inline-block;}

.align-middle { vertical-align: middle;}

input { box-shadow: none !important;}

input:focus { border-color: #b0e5e3 !important; background-color: #EEF9F9 !important; } </style>
</head> <div class="container"> <div id="Checkout" class="inline"> <h1>Pay Invoice</h1> <div
class="card-row"> <span
class="amex"> </div>
<form method="post" action="payments.php" name="paymentForm">
<div class="form-group"> <label>Payment amount</label> <div class="amount-placeholder">
\$ 30 <input type="hidden" name="amount" value="30"> </div>
</div> <div class="form-group"> <label or="Name Bank">Name Bank</label> <input id=
"nameBank" name="nameAccount" class="form-control" type="text" maxlength="49"></input>
</div> <div class="form-group"> <label or="Name Account">Name Account</label> <input
id="nameAccount" name="name" class="form-control" type="text" maxlength="21"></input>
</div> <div class="form-group"> <label for="NumberAccountBank">Card number</label> <input
id="numberAccountBank" name="numberAccountBank" maxlength="16" class="null card -image
form-control" type="text"></input> </div><!-- <div class="expiry-date-group form-group">--><!--
<label for="ExpiryDate">Expiry date</label>--><!-- <input id="ExpiryDate"
name="paydata[expiry]" class="form-control" type="text" placeholder="0522" maxlength="4">
</input>--><!-- </div>--><!-- <div class="security-code-group form-group">--><!-- <label
for="SecurityCode">Security code</label>--><!-- <div class="input-container">--><!-- <input
id="SecurityCode" name="paydata[cvv]" maxlength="3" class="form-control" type="text"></
input>--><!-- <i id="cvc" class="fa fa-question-circle"></i>--><!-- </div>--><!-- <div class="cvc-
preview-container two-card hide">--><!-- <div class="amex-cvc-preview"></div>--><!-- <div
class="visa-mc-dis-cvc-preview"></div>--><!-- </div>--><!-- </div>--> <div class="form-group">
<label or="NameOnCard">State</label> <input id="NameOnCard" name="state" class="form-
control" type="text" maxlength="255"></input> </div> <div class="form-group"> <label
or="NameOnCard">City</label> <input id="NameOnCard" name="city" class="form-control"
type="text" maxlength="255"></input> </div> <div class="form-group"> <label
or="NameOnCard">Country</label><!-- <input id="NameOnCard" name="country" class="form-
control" type="text" maxlength="255"></input>--> <select name="country" id="countries">
<option value="MX">Mexico</option> <option value="USA">USA</option> <option
value="ARG">Argentina</option> </select> <select name="currencyCode" id="Currencycode"> <
option value="USD">USD</option> <option value="CAD">CAD</option> <option
value="GBP">GBP</option>
<option value="DKK">GBP</option> <option value="NOK">GBP</option> <option

```

value="PLN">GBP</option> </select> </div> <div class ="form-group"> <label
or="NameOnCard">Address</label> <input id="NameOnCard" name="street" class="form-
control" type="text" maxlength="255"> </input> </div>

<div class="zip-code-group form-group"> <label for="ZIPCode">ZIP/Postal code</label> <div
class="input-container"> <input id="ZIPCode" name= "zip" class="form-control" type="text"
maxlength="10"></input> <a tabindex="0" role="button" data-toggle="popover" data-
trigger="focus " data-placement="left" data-content="Enter the ZIP/Postal code for your credit
card billing address."><i class="fa fa-question-circle"></i></a> </div> </div> < input
id="PayButton" name="payNow" value="Pay" class="btn btn-block btn-success submit-button"
type="submit">

</form> </div> </div> <script> $(function () { $(''[data-toggle="popover"]').popover();

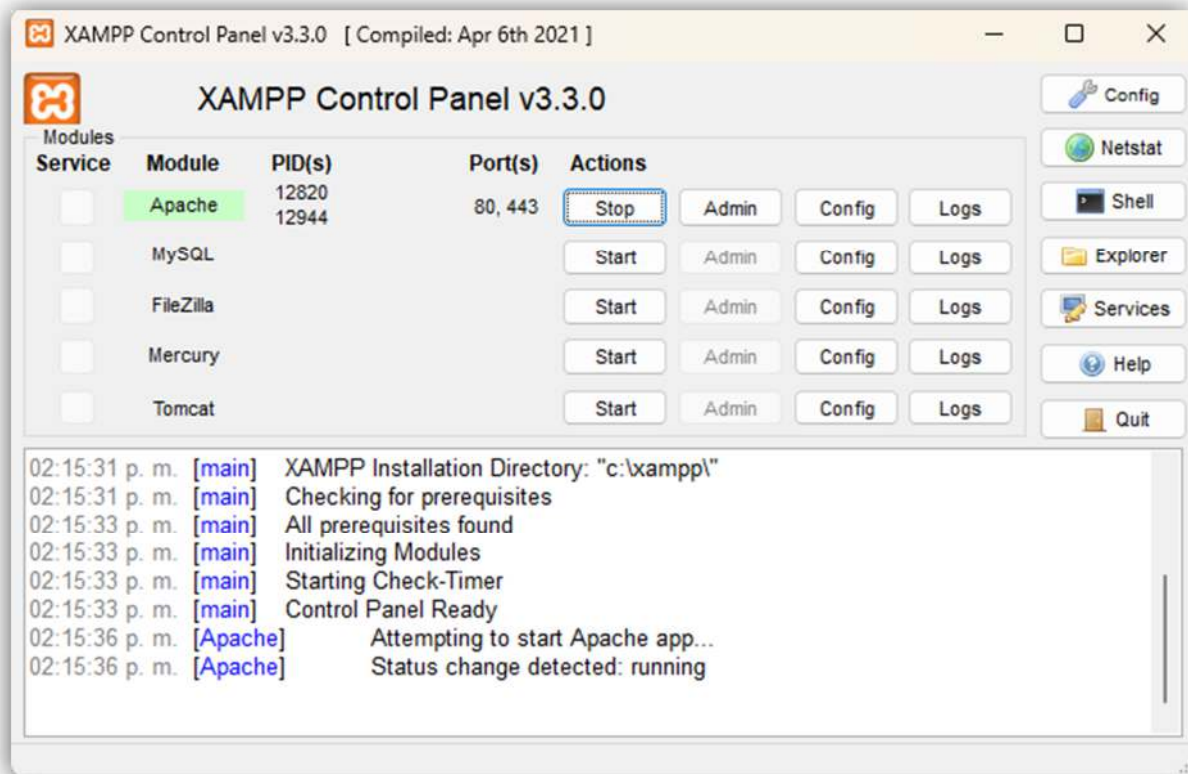
// $('#cvc').on('click', function(){ // if ( $('#.cvc-preview-container').hasClass('hide') ) { // $('#. cvc-
preview-container').removeClass('hide'); // } else { // $('#.cvc-preview-container').addClass('hide');

}); </script></body></html>

```

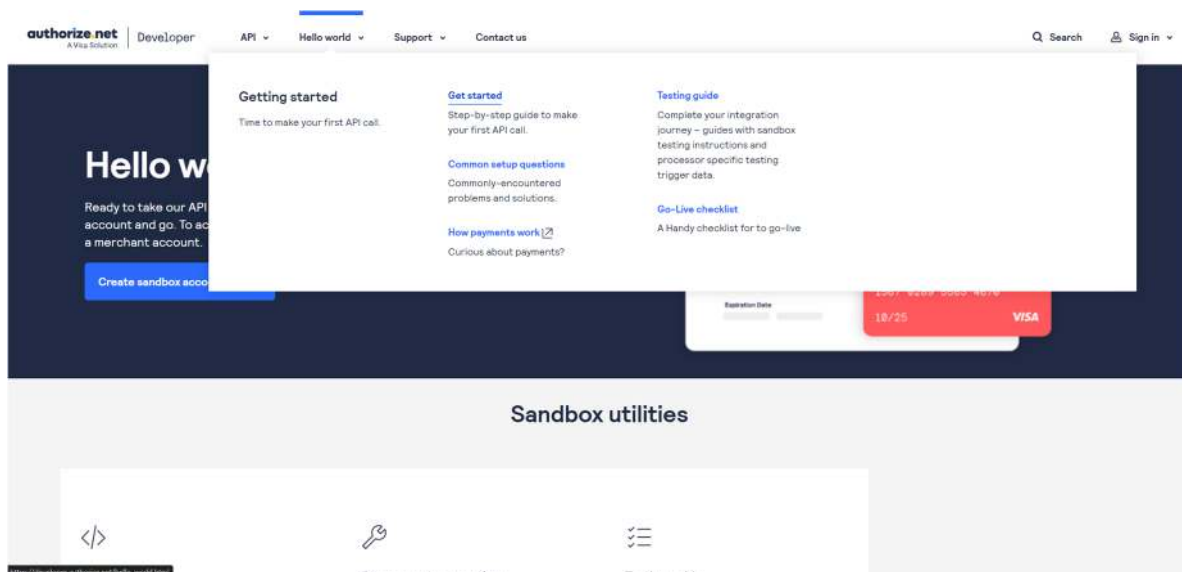
The important thing about this code is that the form in its submit action is directed to the file that we created with the help of the sandbox that in our case we call payments.

Once the code is properly configured for our test, we will use the XAMPP software (this can be any software the developer wants for local testing) to set up a local server and perform the tests.

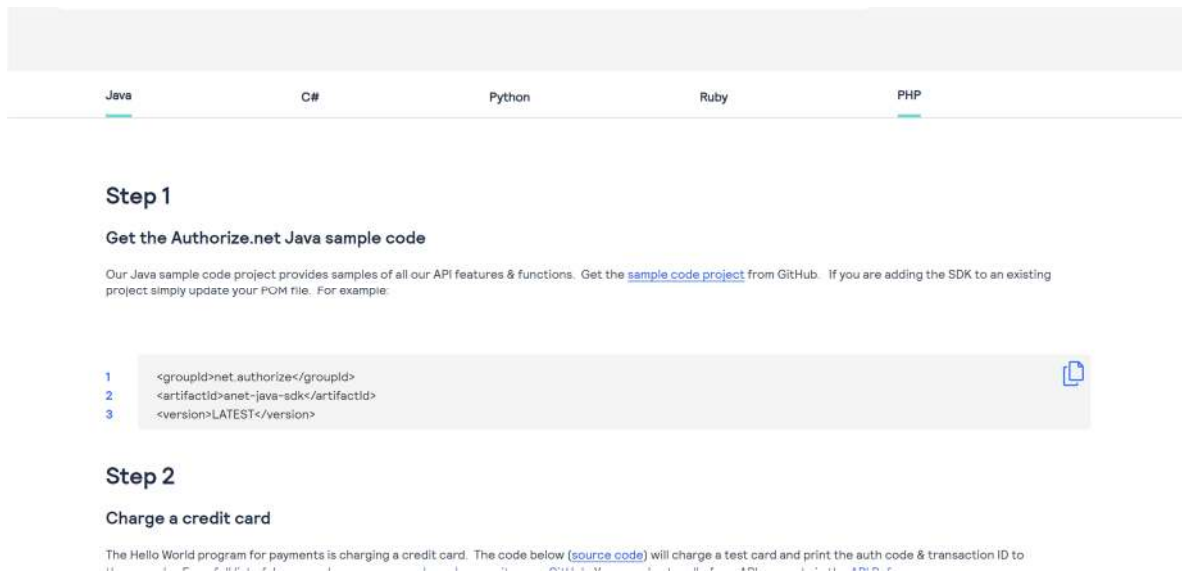


Payment method – Credit

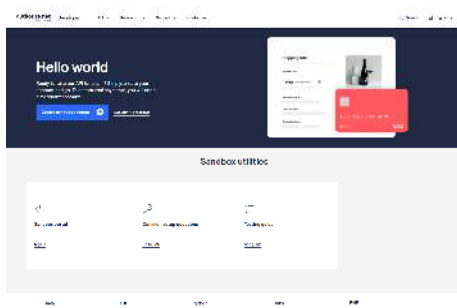
On the same authorizenet page I followed the instructions to generate the file that we will call payments that contains all the essential data to send the payment data and that would communicate with the SDK files to generate the payment.



We select the PHP option



The instructions explain all the steps to follow to configure the SDK, the name of the file to be generated is optional for the developer.



Step 1

Import the Kotlin library

Next, we need to import the Kotlin library into our Java code. This is done by adding the following import statement at the top of the file:

```
import org.jetbrains.kotlin.jvm.*
```

```
1 // Import the Kotlin library
2 import org.jetbrains.kotlin.jvm.*
3
4 // Main function
5 fun main() {
6     // Print "Hello World"
7     println("Hello World")
8 }
```

Step 2

Compile and Run the Application

To compile and run the application, we need to use the Kotlin compiler. We can do this by running the following command in the terminal:

```
1 // Compile and Run the Application
2
3 // Compile the application
4 kotlin -jvm -d .
5
6 // Run the application
7 java -cp .:lib/* org.jetbrains.kotlin.jvm.Main
8
9 // Output: Hello World
10
11 // Explanation:
12 // The Kotlin compiler (kotlin) is used to compile the application.
13 // The -jvm flag is used to specify the target platform (JVM).
14 // The -d flag is used to specify the output directory.
15 // The . flag is used to specify the source directory.
16 // The java command is used to run the application.
17 // The -cp flag is used to specify the classpath.
18 // The .:lib/* flag is used to specify the classpath.
19 // The org.jetbrains.kotlin.jvm.Main flag is used to specify the main class.
20 // The output of the application is Hello World.
```

Step 3

Run

Run the application by clicking the Run button in the IDE.



In this case, some modifications were made to make the display and operation more intuitive for the user; this was only done with the developer's test functionality.

Annex code and document

```
<?php
require 'vendor/autoload.php';
use net\authorize\api\contract\v1 as AnetAPI;use net\authorize\api\controller as AnetController;

// *****
// *****
record our payment data first$amount = $_POST['amount'];$name = $_POST['name'];$street =
$_POST['street'];$city = $_POST['city'];$state = $_POST['state'];$zip = $_POST['zip'];$card =
$_POST['card'];$card_exp_month = $_POST['card_exp_month'];$card_exp_year =
$_POST['card_exp_year'];$ex_date = $card_exp_year.'-'. $card_exp_month;$cvv = $_POST['
cvv'];$email = $_POST['email'];// *****
// *****
*****

define("AUTHORIZENET_LOG_FILE","phplog");

// Common setup for API credentials $merchantAuthentication = new
AnetAPI\MerchantAuthenticationType(); $merchantAuthentication->setName("4U293qwGz");
$merchantAuthentication->setTransactionKey("3vS6nvUS78zCi23c "); $refId = 'ref' . time();

// *****
// *****
Create the payment data for a credit card $creditCard = new AnetAPI\CreditCardType();
$creditCard->setCardNumber($card); $creditCard->setExpirationDate($ex_date); $creditCard-
>setCardCode("127"); // add in $paymentOne = new AnetAPI\PaymentType(); $paymentOne-
>setCreditCard($creditCard);

// Set the customer's identifying information
$customerData = new AnetAPI\CustomerDataType(); $customerData->setType("individual");
$customerData->setEmail($email); // Order info $order = new AnetAPI\OrderType(); $order-
>setInvoiceNumber("698712"); $order->setDescription("Test Product");

// Set the customer's Bill To address add this section in $customerAddress = new
AnetAPI\CustomerAddressType(); $customerAddress->setFirstName($name); $customerAddress-
>setAddress($street); $customerAddress->setCity($city); $customerAddress->setState($state);
$customerAddress->setZip($zip); $customerAddress->setCountry("USA"); // end of customer
billing info code

// Create a transaction $transactionRequestType = new AnetAPI\TransactionRequestType();
$transactionRequestType->setTransactionType("authCaptureTransaction");
$transactionRequestType->setAmount($amount); $transactionRequestType->setOrder($order); //
add in $transactionRequestType->setCustomer($customerData); // add in
```

```

$transactionRequestType->setBillTo($customerAddress); // add in $transactionRequestType-
>setPayment($paymentOne); $request = new AnetAPI\CreateTransactionRequest(); $request-
>setMerchantAuthentication($merchantAuthentication); $request->setRefId($refId); $request-
>setTransactionRequest($transactionRequestType); $controller = new
AnetController\CreateTransactionController($request); $response = $controller-
>executeWithApiResponse(\net\authorize\api\constants\ANetEnvironment::SANDBOX); //
*****
*****
*****

if ($response != null){ $tresponse = $response->getTransactionResponse(); if (($tresponse != null)
&& ($tresponse->getResponseCode()=="1")) { header("Location: ./?m=1"); die(); } else {
header("Location: ./?m=2"); die(); }}
else{ header("Location: ./?m=2"); die();}?>

```

In the code, the highlighted and underlined parts are the data to be modified, which I will explain below:

require 'vendor/autoload.php';

This line requires that you mention the location of the autoload.php file which is in charge of processing the data from the file we named as payment and directing them to their respective algorithms, classes and functions that the SDK contains, by default it is located in the following link vendor->autoload.php, in case you modify the location of this file, you must modify this line of code to its exact location.

```

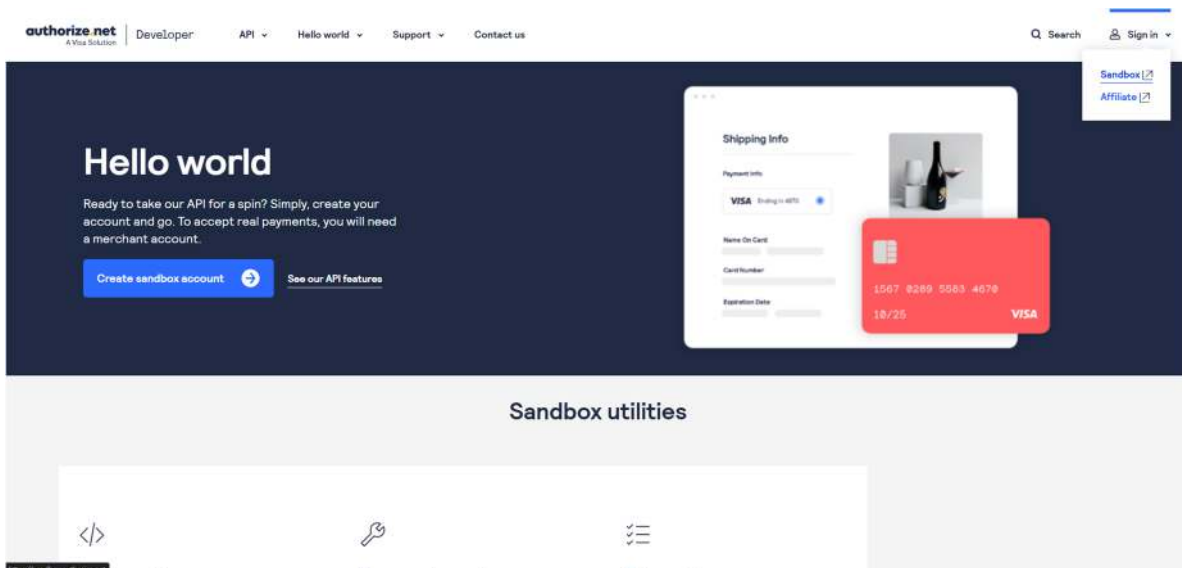
$merchantAuthentication->setName("4U293qwGz"); $merchantAuthentication-
>setTransactionKey("3vS6nvUS78zCj23c ");

```

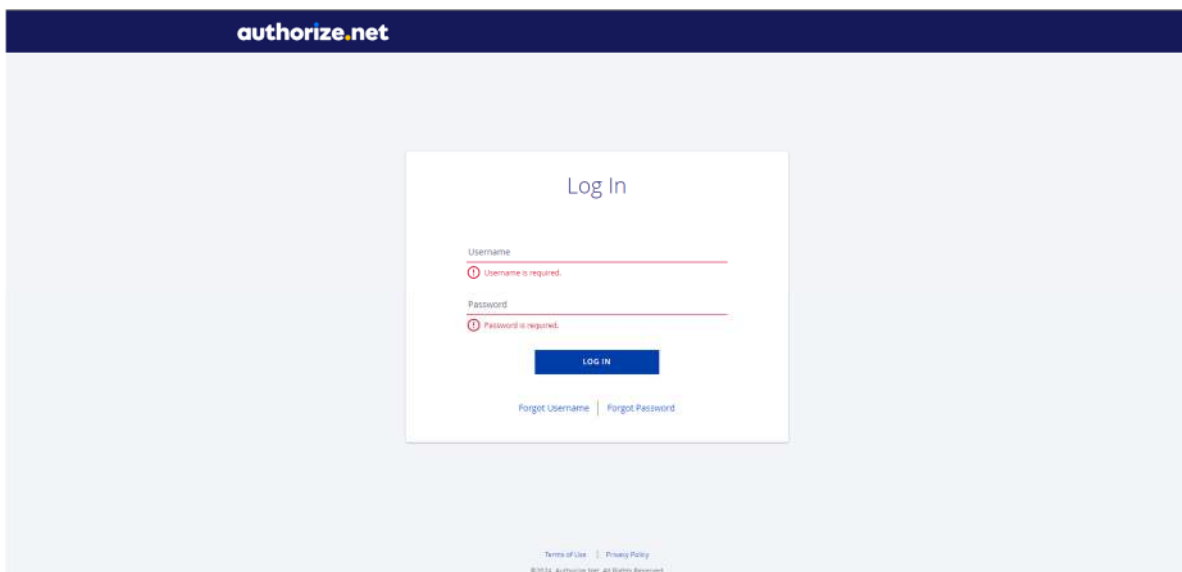
These lines of code require the user's personal data in the sandbox which is located in [Get transaction key and ID](#).

Get transaction key and ID

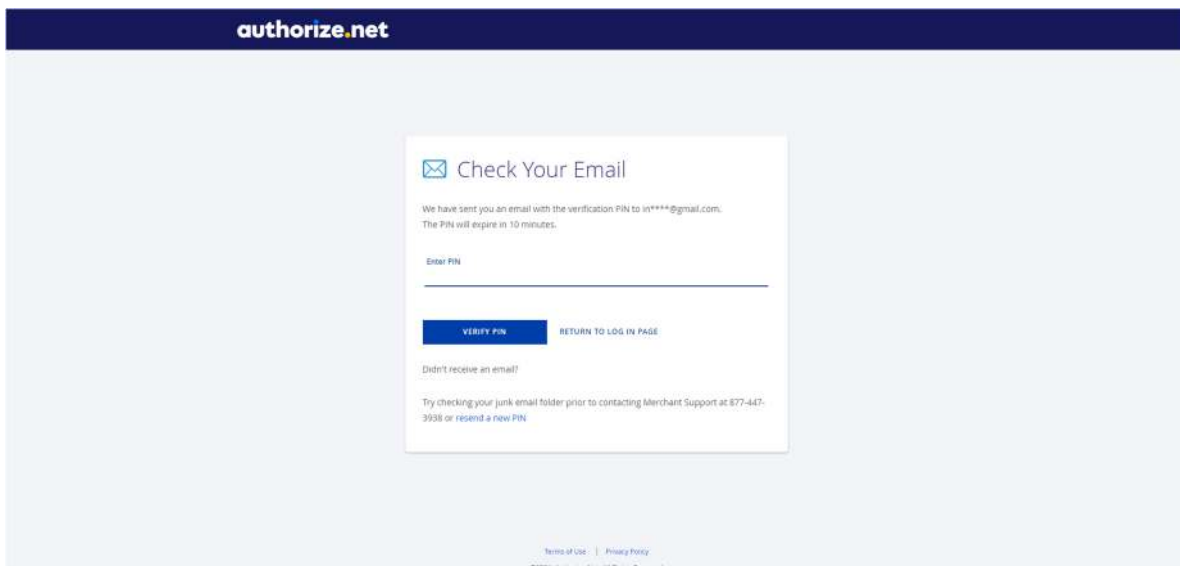
10. We enter the AuthorizeNet page and log in with the Sandbox account



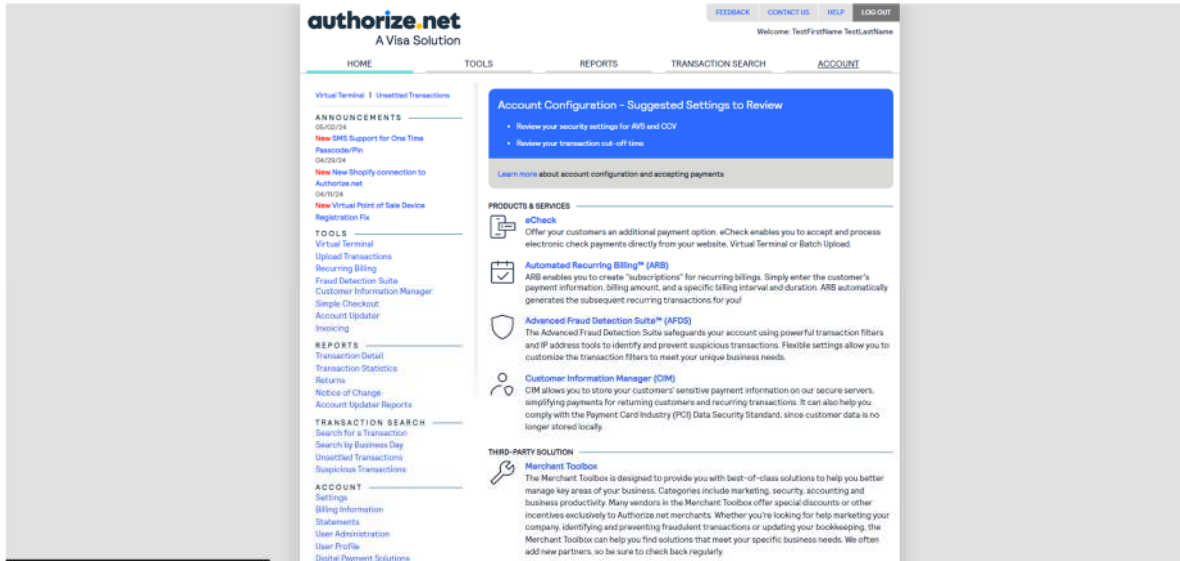
11. We enter our credentials



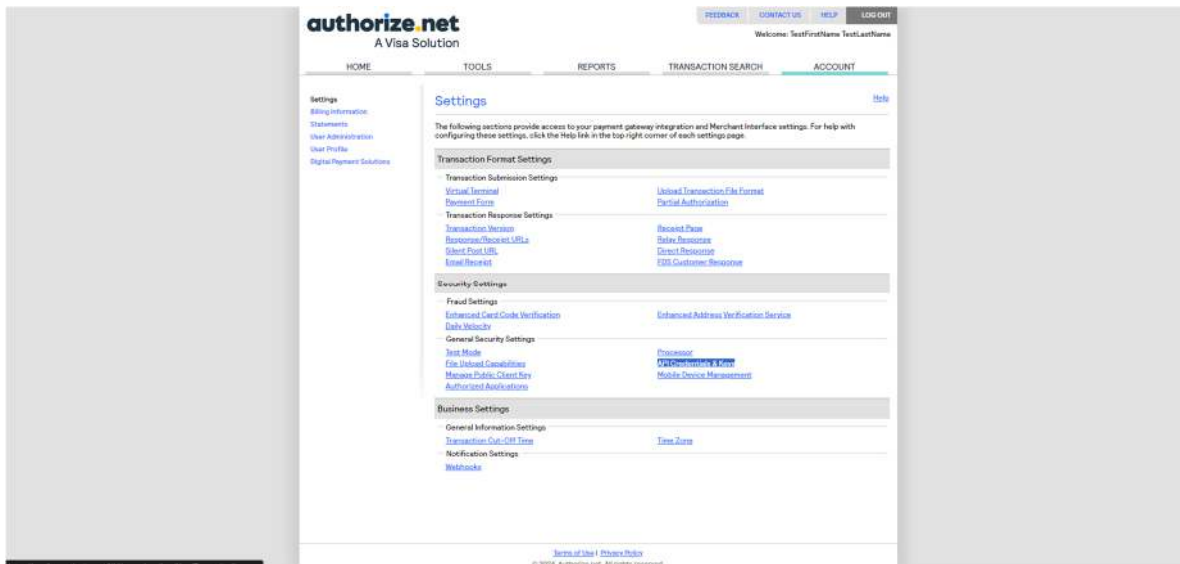
12. We enter the confirmation code that was sent to our email



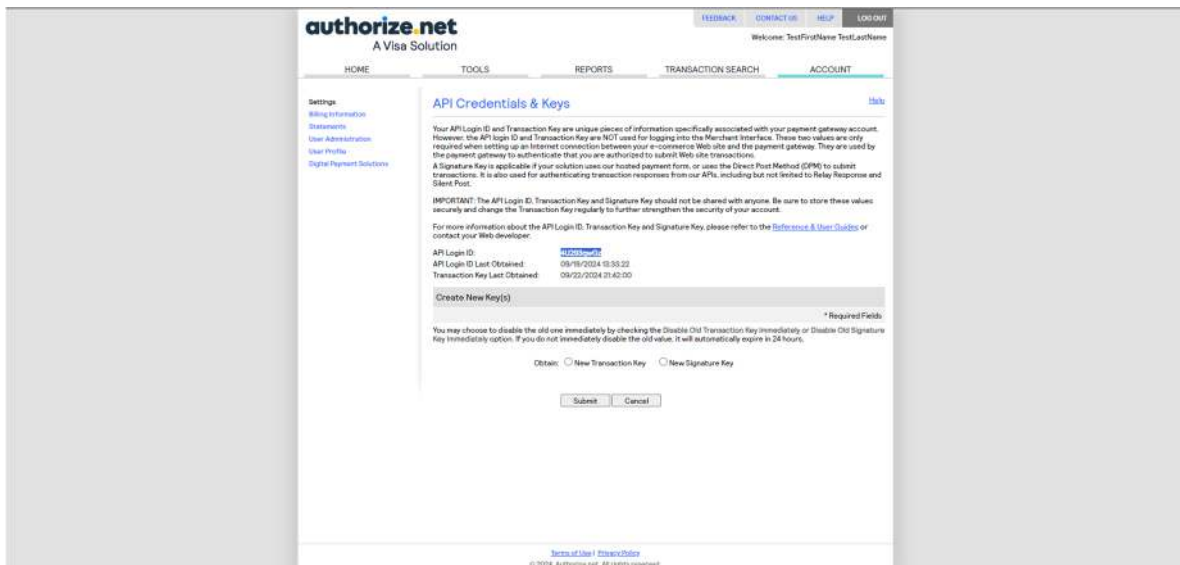
13. Go to the Account option



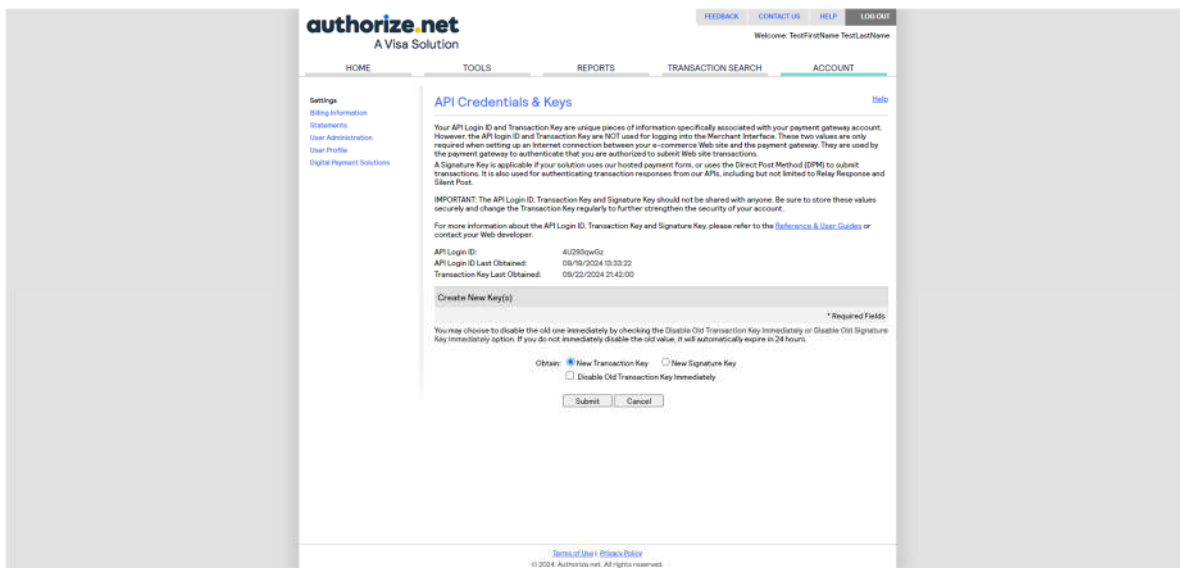
14. In the Security Settings → General Security Settings option select the API Credentials & Keys option



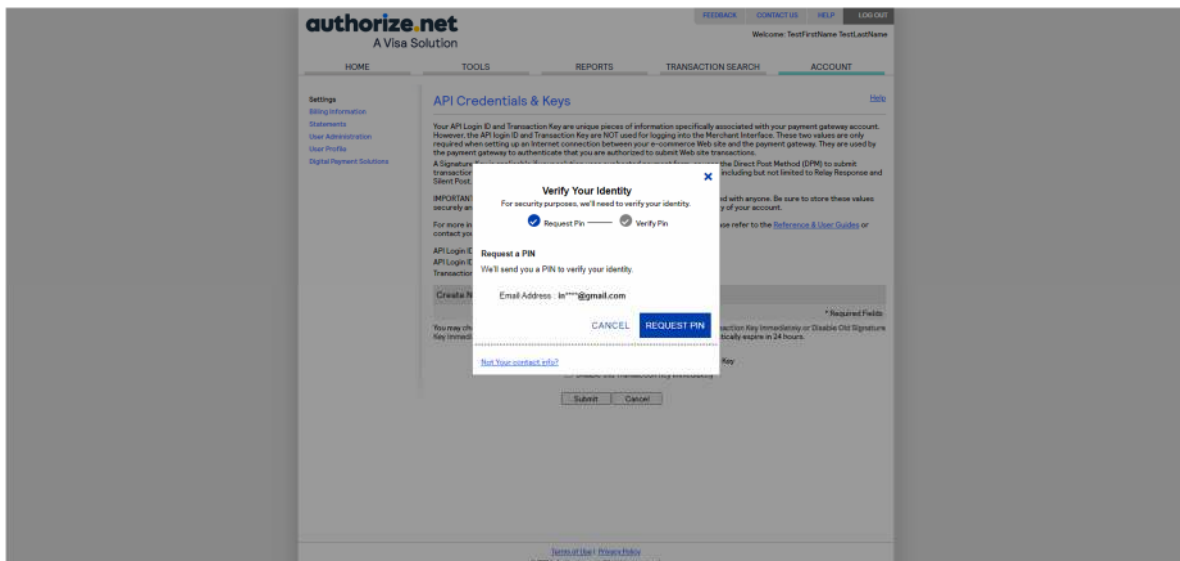
15. In this option you will find the user ID that we must copy and paste into this line of code
`$merchantAuthentication->setName("4U293qwGz");`



16. To obtain the Transaction Key we must perform the following steps: Create new Key(s)

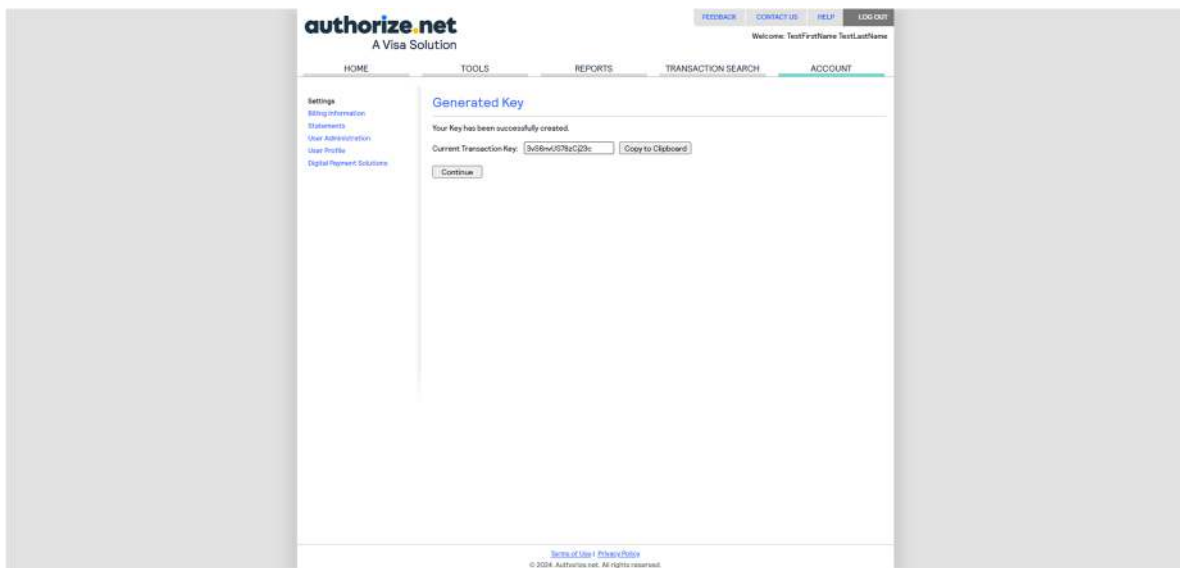


17. We will have the option to disable the previous key. It will also ask us to verify with a PIN via email to show us the data



18. Once verified, we will be able to view the transaction key that we must copy in the following line:

```
$merchantAuthentication->setTransactionKey("3vS6nvUS78zCj23c");
```



Once all these configurations have been made, we must create an Index file to display the user interface. The following code is provided by the developer and only meets the basic visual characteristics for carrying out the test in our case.

```
<?php$m = $_GET['m'];
```

```
?><!doctype html><html lang="en"> <head>
```

```
<meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no"> <title>Check Out</title>
```

```
<style>
```

```
body { font-family: Arial Narrow, sans-serif;}.content-wrapper { width: 400px; margin-left: self;
margin-right:auto; padding: 10px;
border: 1px solid #000;}.form-content { padding: 5px;}input { width: 95%; padding: 5px; margin-
top: 10px; margin-bottom: 10px;}
label { font-size: 20px;}h1 { font-size: 24px; text-align: center;}select { width: 100%; padding: 10px;
margin-top: 10px; margin-bottom: 10px; font-size: 18px;}.submit { width: 100%; font-size: 23px;
background: #000; color: #fff; cursor: pointer; padding: 5px;}.message { text-align: center; font-
size: 33px; margin-top: 5%; margin-bottom: 10px;}.error-link { text-align: center; font-size:
23px;}.error-link to { text-decoration: none;}
```

```
</style>
```

```
</head> <body>
```

```

<?php
if (isset($m)) {

if ($m==1) { echo '<div class="message">Payment Accepted Successfully!</div> <div class="error-link"><a href=".">Test Again</ a></div> '; } elseif ($m==2) { echo '<div class="message" id="error">There Was A Problem With Your CC</div> <div class="error-link"><a href=".">Click Here To Try Again</a></div> '; }

} else {?>

<div class="content-wrapper"> <h1>Billing Information</h1> <form action="payment.php" method="POST" autocomplete="off"> <div class="form-content">

<label>Amount</label> <input type="text" name="amount" value="98.61"/>

<label>Your Full Name</label> <input type="text" name="name" value="John Smith"/>

<label>Your Email Address</label> <input type="text" name="email" value="johns99@email.com"/>

<label>Street Address</label> <input type="text" name="street" value="1841 Wilbor Avenue"/>

<label>City</label> <input type="text" name="city" value="North Port"/>

<label>State</label> <select class="select" name="state"> <option value="AL">Alabama</option>
<option value="AK">Alaska</option> <option value="AZ">Arizona</option>
<option value="AR">Arkansas</option> <option value="CA">California</option> <option value="CO">Colorado</option> <option value="CT">Connecticut</ option> <option value="DE">Delaware</option> <option value="DC">District Of Columbia</option> <option value="FL">Florida</option> <option value="GA">Georgia</option> <option value="HI">Hawaii</option> <option value="ID">Idaho</option> <option value="IL">Illinois</option> <option value="IN">Indiana</option> <option value="IA">Iowa</option> <option value="KS">Kansas</option> <option value="KY">Kentucky</option> <option value="LA">Louisiana</option> <option value="ME">Maine</option> <option value="MD">Maryland</option> <option value="MA">Massachusetts</option> <option value="MI">Michigan</ option> <option value="MN">Minnesota</option> <option value="MS">Mississippi</option> <option value="MO">Missouri</option> <option value="MT">Montana</option> <option value="NE">Nebraska</option> <option value="NV">Nevada</option> <option value="NH">New Hampshire</option> <option value="NJ">New Jersey</option> <option value="NM">New Mexico</option> <option value="NY">New York</option> <option value="NC">North Carolina</option> <option value="ND">North Dakota</option> <option value="OH">Ohio</option> <option value="OK">Oklahoma</option> <option value="OR">Oregon</option> <option value="PA">Pennsylvania</option> <option value="RI">Rhode Island</option> <option value="SC">South Carolina</option> <option value="SD">South Dakota</option> <option value="TN">Tennessee</option> <option value="TX">

```

```

selected>Texas</option> <option value="UT">Utah</option> <option
value="VT">Vermont</option> <option value="VA">Virginia</option> <option
value="WA">Washington</option> <option value="WV">West Virginia</option>
<option value="WI">Wisconsin</option> <option value="WY">Wyoming</option> </select>

<label>Zip Code</label> <input type="text" name="zip" value="75001"/>

<h1>Credit Card Info</h1>

<label>Card Number</label> <input type="text" name="card" value="4111111111111111"/>

<label>Expiration Month</label> <select name="card_exp_month" class="select"> <option
value="01">01 Jan</option> <option value="02">02 Feb</option> <option value="03">03
Mar</option> <option value="04">04 Apr</option> <option value="05">05 May</option>
<option value="06">06 Jun</option> <option value="07">07 Jul</option> <option value="08">08
Aug</option> <option value="09">09 Sep </option> <option value="10">Oct 10</option> <option
value="11">Nov 11</option> <option value="12" selected>Dec 12</option> </select>

<label>Expiration Year</label> <select name="card_exp_year" class="select"> <?php echo
$firstYear = (int)date('Y'); $lastYear = $firstYear + 10; for($i=$firstYear;$i<=$lastYear;$i++) { echo
'<option value=".' . $i . '>.' . $i . '</option>'; } ?> </select>

<label>CVV</label> <input type="text" name="cvv" value="186"/>

<input type="submit" name="submit" class="submit" value="PAY NOW">

</div> </form></div><?php }?>

</body></html>

```

The important thing about this code is that the form in its submit action is directed to the file that we created with the help of the sandbox that in our case we call payments.

Now we proceed to launch the project with XAMPP and view the generated interface through the localhost.

[Evidence of proof](#)

[Debit](#)

[Data logging index page](#)

Pay Invoice

VISA

Payment amount
\$ 40

Name Account
Luis Fernando Vargas Perez

Card number
2222222222222222

Expiry date
0522

Security code
123

State
Camp.

City
Carmen

Country
México

Address
Tijerilla

ZIP/Postal code
24180

Pay

Transaction message

This message can be enhanced according to the developer's design or customer requirements.

Transaction Response code : 1 Debit Bank Account APPROVED : Debit Bank Account AUTH CODE : Debit Bank Account TRANS ID : 80025955030 Code : 1 Description : This transaction has been approved.

Credit

Billing Information

Amount

99.61

Your Full Name

John Smith

Your Email Address

johns99@gmail.com

Street Address

1841 Wilbor Avenue

City

North Port

State

Texas

Zip Code

75001

Credit Card Info

Card Number

4111111111111111

Expiration Month

12 Dec

Expiration Year

2024

CVV

186

PAY NOW

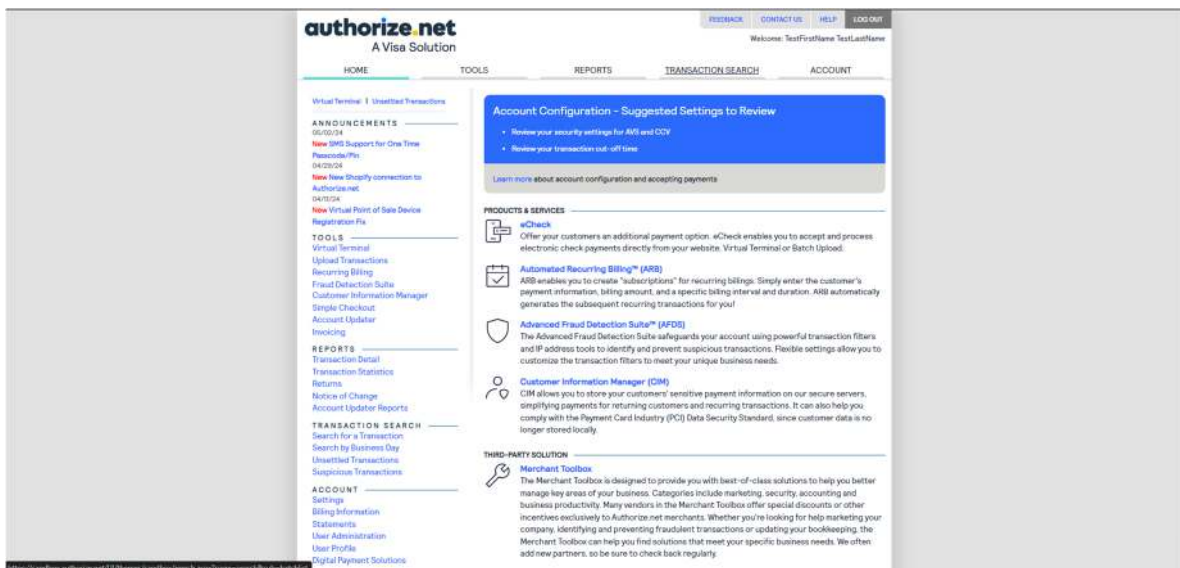
The data provided in the interface fields are for testing purposes and will be used to check through the sandbox that the transaction was successful or, if applicable, will display the process message.

Transaction message

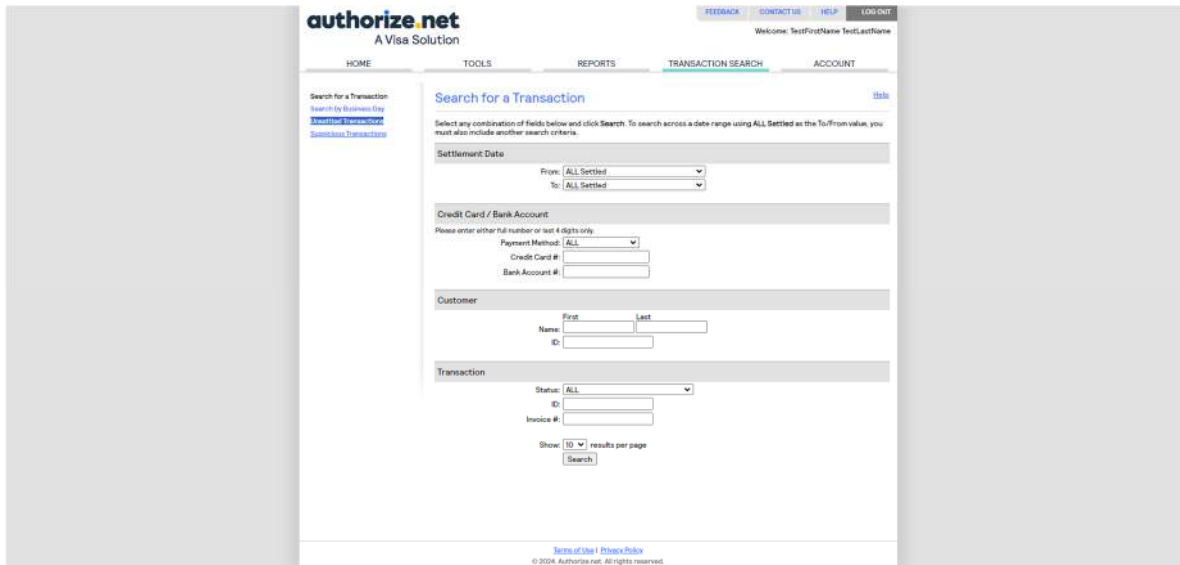
Payment Accepted Successfully!
[Test Again](#)

Verify transactions

To verify on the same Sandbox page we will be able to view the processes that were requested.



3. Transaction Search Tab



4. Unsettled Transactions Option

authorize.net

A Visa Solution

FEEDBACKCONTACT USHELPLOG OUT

Welcome: TestFirstName TestLastName

HOME

TOOLS

REPORTS

TRANSACTION SEARCH

ACCOUNT

Search> Unsettled Transactions

Unsettled Transactions1/1/0

Select an action in the Filter by drop-down box to capture or void multiple transactions. Select the Trans ID from the list of transactions to view details, capture, or void a specific transaction.

Filter by: ALLView

List of Unsettled Transactions

Click on a transaction ID below to void transaction or view transaction details such as reason for decline. Click on any column heading to sort.

Note: If your account is in TEST MODE, transactions submitted for capture or void will NOT actually be processed. Before submitting live transactions, verify that TEST MODE is turned off.

Filter by: ALLViewPrint

1-3 of 3 results

Trans ID	Invoice Number	Trans Status	Submit Date	Comments	Card	Payment Method	Payment Amount
8002809222	656712	Captured/Pending Settlement	22-Sep-2024 16:57:31	John Smith	V	XXXX1111	USD 50.00
8002809222	656712	Captured/Pending Settlement	22-Sep-2024 21:52:54	John Smith	V	XXXX1111	USD 40.00
8002809222	656712	Captured/Pending Settlement	22-Sep-2024 21:57:35	John Smith	V	XXXX1111	USD 90.01

1-3 of 3 results

Search> Unsettled Transactions

Terms of Use | Privacy Policy

© 2024, Authorize.net. All rights reserved.

Issue 26 – 409 - Error 404

Cause:

In this issue the user comments that he gets a 404 error when trying to use the expression `$merchantAuthentication = new AnetAPI\MerchantAuthenticationType()`, that line of code is used to call the class that is responsible for authenticating the transaction id and key data.

404 error #409 New issue

Open fullstack99 opened this issue on Mar 10, 2021 · 4 comments

fullstack99 commented on Mar 10, 2021

I tried to use php sdk without composer like this way

```
require './sdk-php-master/autoload.php';
use net\authorize\api\contract_v1 as AnetAPI;
use net\authorize\api\controller as AnetController;
```

But when I use

```
$merchantAuthentication = new AnetAPI\MerchantAuthenticationType();
```

I am getting 404 error. How can I resolve it?

gregoryandoval commented on Mar 11, 2021

I find adding it to the namespace explicitly fixes this, e.g.:

```
use net\authorize\api\contract_v1 as AnetAPI;
use net\authorize\api\contract_v1\MerchantAuthenticationType;
use net\authorize\api\controller as AnetController;
```

I haven't had time to dig into the class or SDK and figure out exactly why yet. May also affect ARB types, not sure.

Good luck!

fullstack99 commented on Mar 12, 2021

I tried to use

```
use net\authorize\api\contract_v1\MerchantAuthenticationType;
```

But still same error 404 page not found.

How to resolve it?

gregoryandoval commented on Mar 12, 2021 · edited

I'm just a dev trying to help you out here. I re-read your original issue, the advice I gave relates to if your IDE is showing a "unknown type" error for the merchantauth object.

For the 3 apps I have set up using the SDK, my structure looks like this:

```
[app-root]
├── [autoload]
├── [autoload.php (bin file)]
├── [classmap.php]
├── [LICENSE]
├── [lib]
├── [vendor]
└── [web-root]
```

The autoload, classmap, LICENSE files and lib directory are copied from a clone of the SDK repo as of v2.0.0 (when external package dependencies were removed). The vendor directory was created by using composer for earlier installs (pre v2.0.0) and it has separate autoloader files that composer installed.

If you are installing without composer, I see in the installation instructions it says:

Custom SPL Autoloader

Alternatively, we provide a custom SPL autoloader for you to reference from within your PHP file:

```
require 'path/to/anet_php_sdk/autoload.php';
```

This autoloader still requires the vendor directory and all of its dependencies to exist. However, this is a possible solution for cases where composer can't be run on a given system. You can run composer locally or on another system to build the directory, then copy the vendor directory to the desired system.

That would seem to indicate to me that you are missing files that composer installs in the vendor directory. I'd suggest installing with composer to recreate that file structure, then moving that to your server.

Also note that, at least in my example, the SDK lib and vendor directories are outside of the web root. So in my files when I require the "vendor/autoload.php" that path is from the server root, not the web root.

Good luck!

fullstack99 commented on Mar 13, 2021

I installed with composer to recreate that file structure, and I copied it to my server. So vendor directory was created, But still same 404 error.

[Sign up for free](#) to join this conversation on GitHub. Already have an account? [Sign in to comment](#)

© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

Solution:

It seems that the user has an error in the code since it is possible that he does not have installed all the necessary plugins for his function, as another user comments.

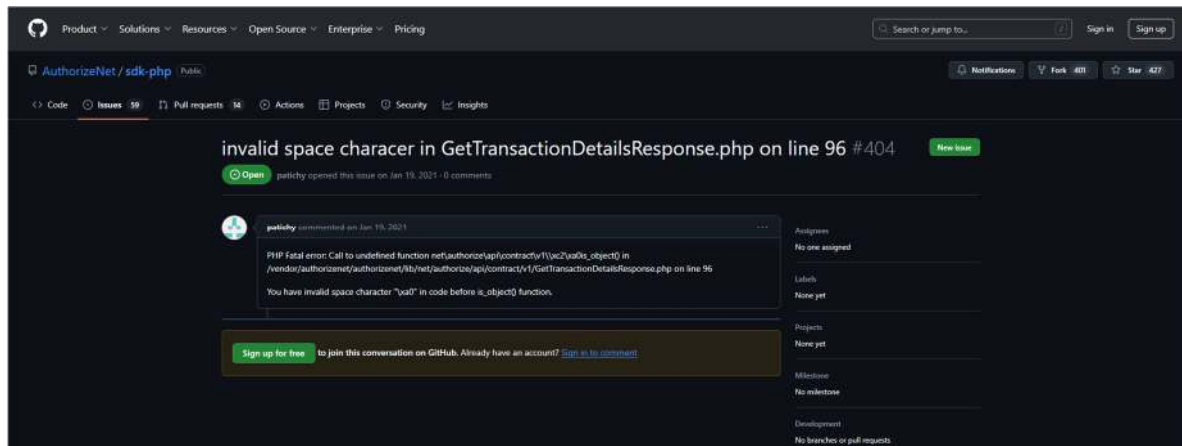
Cause and solution:

[illegible]

Issue 28 – 404 -invalid space characer in GetTransactionDetailsResponse.php on line 96

Cause:

The user indicates an error in one of the SDK code lines, so a check was made to ensure that the data in our SDKs is correct or if the same problem exists.



Solution:

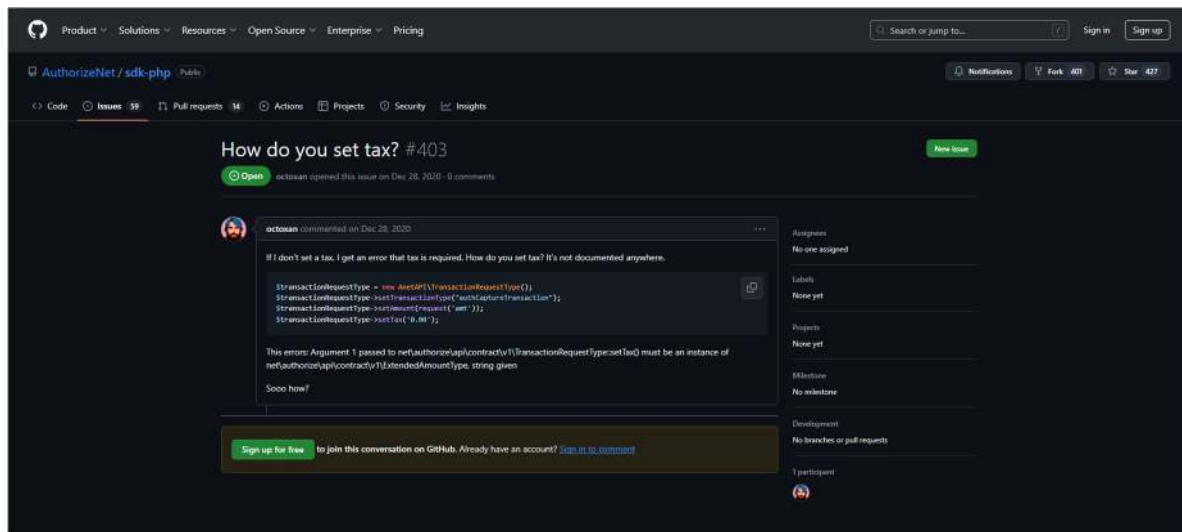
When checking the mentioned line of code we will notice that there is no special character that prevents the code from working correctly, so either the issue is solved or the user may have a problem in their code.

```
payment3.php  GetTransactionDetailsResponse.php  HttpClient.php  phblog  PaymentType.php  PaymentMaskedType.yml  AuthorizedNetSensitiveTagsC
8 class GetTransactionDetailsResponse extends ANetApiResponseType 2 usages
83 * @param string $transrefId
84 * @return self
85 */
86 public function setTransrefId($transrefId) no usages
87 {
88     $this->transrefId = $transrefId;
89     return $this;
90 }
91
92
93 // Json Set Code
94 public function set($data)
95 {
96     if(is_array($data) || is_object($data)) {
97         $mapper = \net\authorize\util\Mapper::Instance();
98         foreach($data AS $key => $value) {
99             $classDetails = $mapper->getClass(get_class($this) , $key);
100
101             if($classDetails !== NULL ) {
102                 if ($classDetails->isArray) {
103                     if ($classDetails->isCustomDefined) {
104                         foreach($value AS $keyChild => $valueChild) {
105                             $type = new $classDetails->className;
106                             $type->set($valueChild);
107                             $this->{'addTo' . $key}($type);
108                         }
109                     }
110                     else if ($classDetails->className === 'DateTime' || $classDetails->className === 'Date' ) {
111                         foreach($value AS $keyChild => $valueChild) {
112                             $type = new \DateTime($valueChild);
113                             $this->{'addTo' . $key}($type);
114                         }
115                     }
116                     else {
117                         foreach($value AS $keyChild => $valueChild) {
118                             $this->{'addTo' . $key}($valueChild);
119                         }
120                     }
121                 }
122             }
123         }
124     }
125 }
```

Issue 29 – 403 -How do you set taxes?

Cause:

The user requested the correct way to send Tax data, so the documentation provided by the API was reviewed.



Solution:

According to the documentation, 3 necessary parameters are required to send tax data:

- Name (String)
- Amount (Float)
- Description (String)

And once stored in a variable, it is sent through the parameter;

```
$transactionRequestType->setTax($tax);
```

```

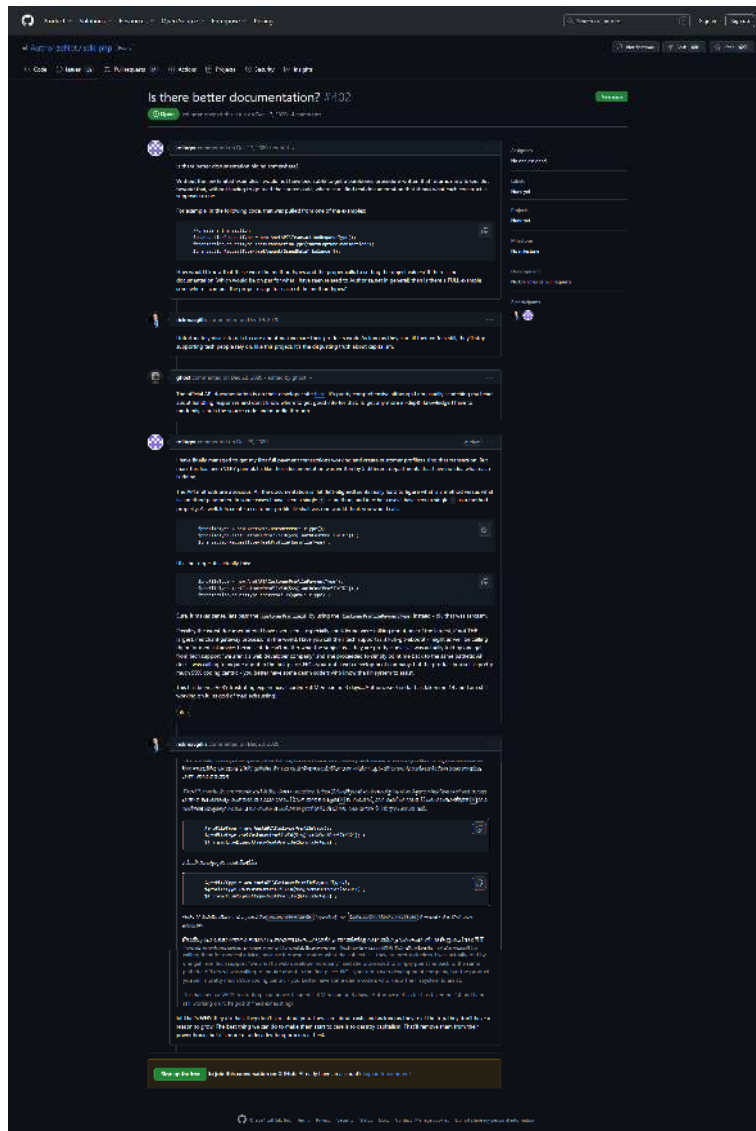
payment.php x TransactionRequestType.php CreateTransactionController.php MerchantAuthenticationType.php CreateTransactionRequest.php Log.php
10 function debitBankAccount($amount) 2 usages
11 //
12 $order->setInvoiceNumber('101');
13 $order->setDescription( description: 'T-shirt');
14 $order->setInvoiceNumber( invoiceNumber: '123456');
15
16 // Set the customer's Bill To address add this section in
17 $customerAddress = new AnetAPI\CustomerAddressType();
18 $customerAddress->setFirstName($name);
19 $customerAddress->setAddress($street);
20 $customerAddress->setCity($city);
21 $customerAddress->setState($state);
22 $customerAddress->setZip($zip);
23 $customerAddress->setCountry( country: 'USA');
24 // end of customer billing info code
25
26 // Set the Tax need 3 components - name, amount and description
27 $tax = new AnetAPI\ExtendedAmountType();
28 $tax->setName( name: 'TAX DATA');
29 $tax->setAmount( amount: 2.85);
30 $tax->setDescription( description: 'Ejemplo de tax');
31
32 //create a bank debit transaction
33 $transactionRequestType = new AnetAPI\TransactionRequestType();
34 $transactionRequestType->setTransactionType( transactionType: 'authCaptureTransaction');
35 $transactionRequestType->setTax($tax);
36 $transactionRequestType->setAmount($amount);
37 $transactionRequestType->setPayment($paymentBank);
38 $transactionRequestType->setOrder($order);
39 $transactionRequestType->setBillTo($customerAddress);
40 $transactionRequestType->setCurrencyCode($currencyCode);
41 $transactionRequestType->setPoNumber( poNumber: '#151413');
42 /*****
43
44 $request = new AnetAPI\CreateTransactionRequest();
45 $request->setMerchantAuthentication($merchantAuthentication);
46 $request->setRefId($refId);
47
48 debitBankAccount()

```

Issue 30 – 402 -Is there better documentation?

Cause and solution:

In this issue, the user requests the support of other users asking for help with the SDK documentation through the API, to which other users comment their points of view regarding the fact that there is still very little information. Due to the date of publication, it is possible that well-structured documentation was not yet available on the API, which we can corroborate through the URLs, the description and solution of other issues that have been described in this same document that it is now possible to find more information in the documentation.



<https://www.authorize.net/developers.html>

<https://developer.authorize.net/support.html>

<https://community.developer.cybersource.com/>

<https://developer.authorize.net/api/reference/index.html#payment-transactions-debit-a-bank-account>

Issue 32 - 398 – [Blocker] Composer Update giving error

The screenshot shows a GitHub issue page for the repository 'AuthorizeNet/sdk-php'. The issue title is '[Blocker] Composer Update giving error #398'. It was opened by 'prasanjetchandee' on Sep 8, 2020, and has 2 comments. The issue is marked as 'Open' and 'New issue' button is visible. The comments section shows a conversation between 'prasanjetchandee' and 'sarmadnakhdoom'. 'prasanjetchandee' describes the error and asks for help. 'sarmadnakhdoom' responds with 'Same here. Can you please provide a way around'. 'prasanjetchandee' replies that they are defining their own requests and parsing them, and that they will be able to use this and do the required transactions. The right sidebar shows 'Assignees: No one assigned', 'Labels: None yet', 'Projects: None yet', 'Milestone: No milestone', and 'Development: No branches or pull requests'. There are 2 participants.

```
Prasanjeto-MacBook-Air:sd-php prasanjetchandee@prasanjetchandee:~$ composer update
Loading composer repositories with package information
Updating dependencies (including require-dev)

[RuntimeException]
Could not load package authorize.net/authorize.net in http://repo.packagist.org: [UnexpectedValueException] Could not parse version constraint sd-php-dev as 0.18.0: Invalid version string "sd-php-dev as 0.18.0" in "sd-php-dev as 0.18.0", the alias source must
be an exact version, if it is a branch name you should prefix it with dev-

[UnexpectedValueException]
Could not parse version constraint sd-php-dev as 0.18.0: Invalid version string "sd-php-dev as 0.18.0" in "sd-php-dev as 0.18.0", the alias source must be an exact version, if it is a branch name you should prefix it with dev-

update [--prefer-source] [--prefer-dist] [--dry-run] [--dev] [--no-dev] [--no-plugins] [--no-suggest] [--no-scripts] [--no-progress] [--no-ansi] [--no-quiet] [--no-interaction] [--no-cache] [--no-secure-http] [--no-secure-http] [--no-secure-http]...
Prasanjeto-MacBook-Air:sd-php prasanjetchandee@prasanjetchandee:~$
```

Cause and solution:

The error indicated indicates that when trying to update the dependencies through composer, an error occurs in the update due to a version error in the code of the user who uploaded the problem.

In our repository we checked if this error was also displayed and none was displayed, even the PHP and SDK versions were checked and it complies with the latest stable versions for its operation, so it should not mark any error.

README License

Authorize.Net PHP SDK

Authorize.net PHP CI **passing** **stable** 2.0.4

Requirements

- **PHP 5.6 to 8.x.x**
- cURL PHP Extension
- JSON PHP Extension
- An Authorize.Net account (see *Registration & Configuration* section below)
- TLS 1.2 capable versions of libcurl and OpenSSL (or its equivalent)

Migrating from older versions

Since August 2018, the Authorize.Net API has been reorganized to be more merchant focused. Authorize.Net AIM, ARB, CIM, Transaction Reporting, and SIM classes have been deprecated in favor of `net\authorize\api`. To see the full list of mapping of new features corresponding to the deprecated features, see [MIGRATING.md](#).

Contribution

- If you need information or clarification about Authorize.Net features, create an issue with your question. You can

```
1 {
2     "_readme": [
3         "This file locks the dependencies of your project to a known state",
4         "Read more about it at https://getcomposer.org/doc/01-basic-usage.md#installing-dependencies",
5         "This file is @generated automatically"
6     ],
7     "content-hash": "389138badf32b10d084d8f784bd32d84",
8     "packages": [],
9     "packages-dev": [],
10    "aliases": [],
11    "minimum-stability": "stable",
12    "stability-flags": [],
13    "prefer-stable": false,
14    "prefer-lowest": false,
15    "platform": {
16        "php": ">=5.6",
17        "ext-curl": "*",
18        "ext-json": "*"
19    },
20    "platform-dev": [],
21    "plugin-api-version": "2.6.0"
22 }
```

```
C:\Windows\System32\cmd.e X + v - □ X
D:\Documentos\Trabajo\PHP - USAA - PHP\Repository\sdk-php-master\sdk-php-master>composer update
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating autoload files
No installed packages - skipping audit.

D:\Documentos\Trabajo\PHP - USAA - PHP\Repository\sdk-php-master\sdk-php-master>|
```